

Programski prevodioci 1

- Vežbe VIII -

|| teams.microsoft.com is sharing your screen. [Stop sharing](#) [Hide](#)

+7

BT

AH

MK

УС

AK

СЧ

JM

KM

PB

НБ

ФС

Ана Кнежевић

София Чоловић

Јован Мильковић

Катарина Милошевић

Радивоје Васиљевић

Никола Болесников

Филип Ступар

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Moguće vrednosti:
Con, Var, Type,
Meth, Fld, Elem,
Prog

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals



Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Tip se opisuje strukturmim čvorom.

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals



Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Za metode adresa na kojoj se nalaze. Za promenljive i polja klase njihov redni broj.

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Za promenljive njihov nivo ugnezđavanja (lokalne, globalne). Za metode broj formalnih argumenata.

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Za program kolekcija simbola programa. Za metode kolekcija lokalnih promenljivih i argumenata

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Strukturni čvorovi

Čuvaju informacije o strukturama tipa.

kind
elementType
n
fields

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Strukturni čvorovi

Čuvaju informacije o strukturama tipa.

- kind
- elementType
- n
- fields

Moguće vrednosti:
None, Int, Char, Array, Bool, Class

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Strukturni čvorovi

Čuvaju informacije o strukturama tipa.

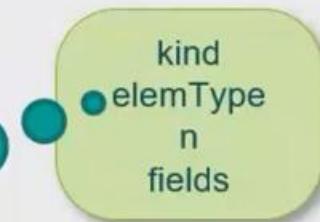


Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

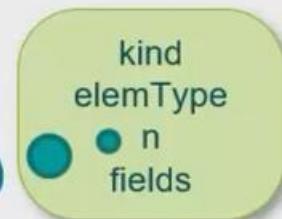
Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Strukturni čvorovi

Čuvaju informacije o strukturama tipa.



Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

Strukturni čvorovi

Čuvaju informacije o strukturama tipa.



Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

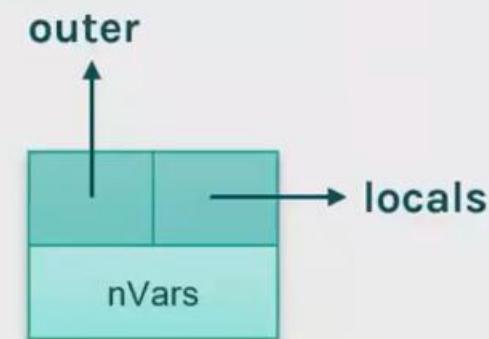
Strukturni čvorovi

Čuvaju informacije o strukturama tipa.



Čvorovi opsega

Služe za manipulaciju opsezima važenja imena.



Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

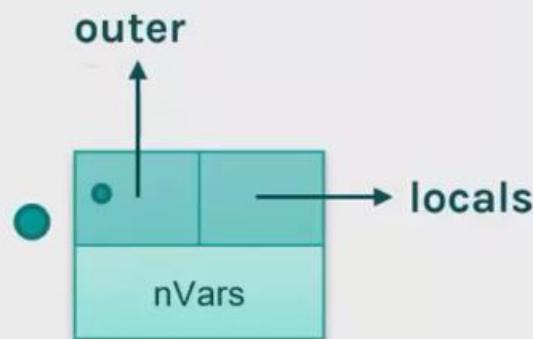
Strukturni čvorovi

Čuvaju informacije o strukturama tipa.



Čvorovi opsega

Služe za manipulaciju opsezima važenja imena.



Pokazivač na okružujući opseg.

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

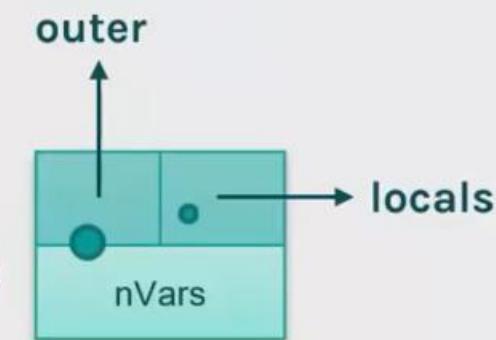
Strukturni čvorovi

Čuvaju informacije o strukturama tipa.



Čvorovi opsega

Služe za manipulaciju opsezima važenja imena.



Pokazivač na listu imena deklarisanih unutar opsega.

Programski prevodioci 1

Tabela simbola

Tabela simbola za jezik Mikrojava ima 3 vrste čvorova:

Objektni čvorovi

Čuvaju informacije o deklarisanim imenima.

kind
name
type
next
adr
level
locals

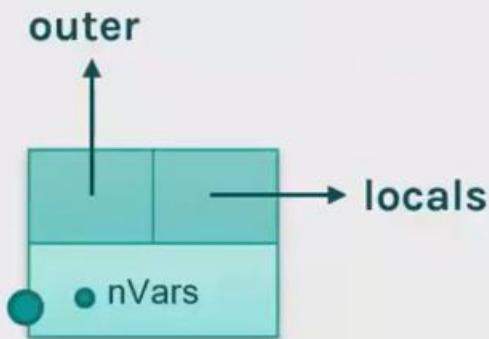
Strukturni čvorovi

Čuvaju informacije o strukturama tipa.



Čvorovi opsega

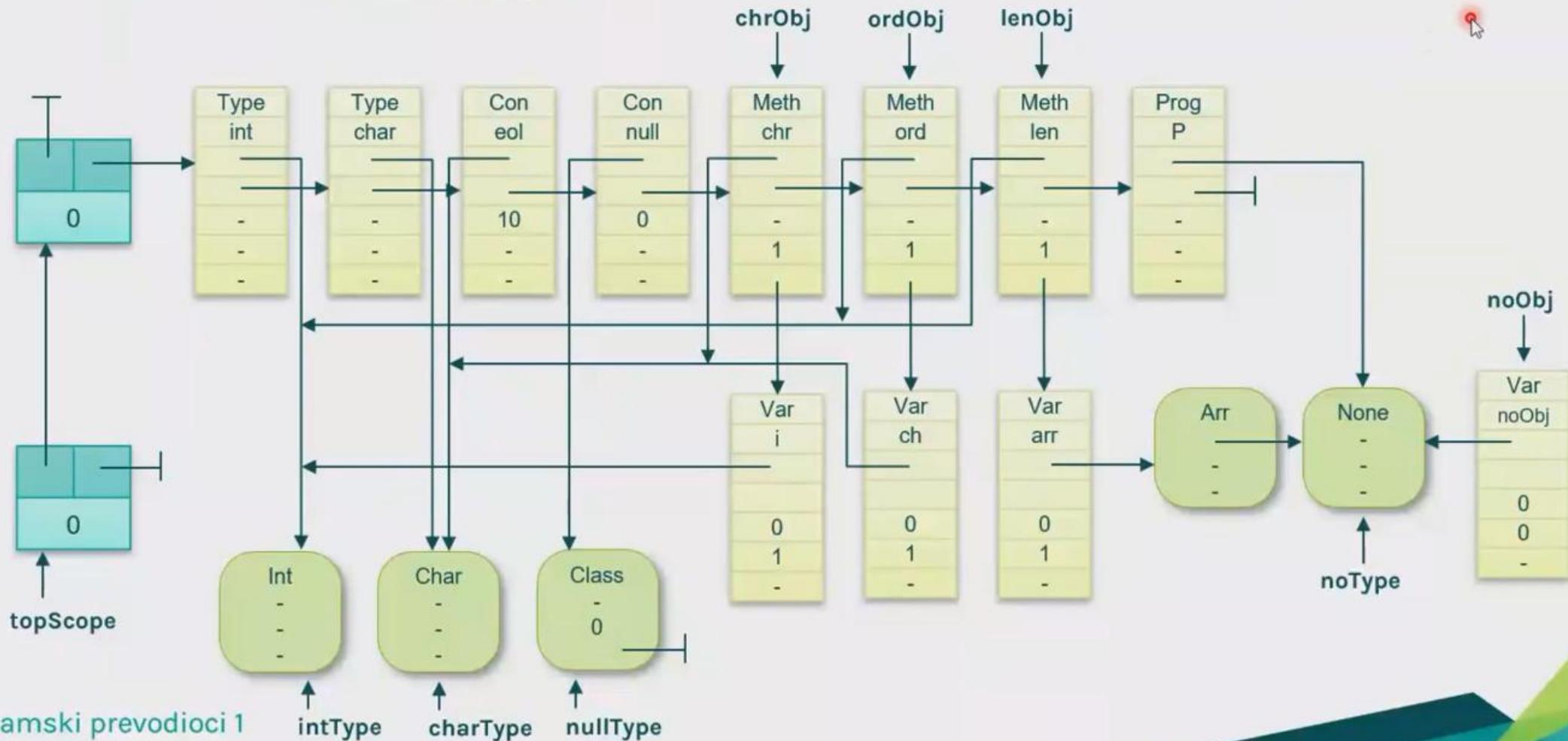
Služe za manipulaciju opsezima važenja imena.



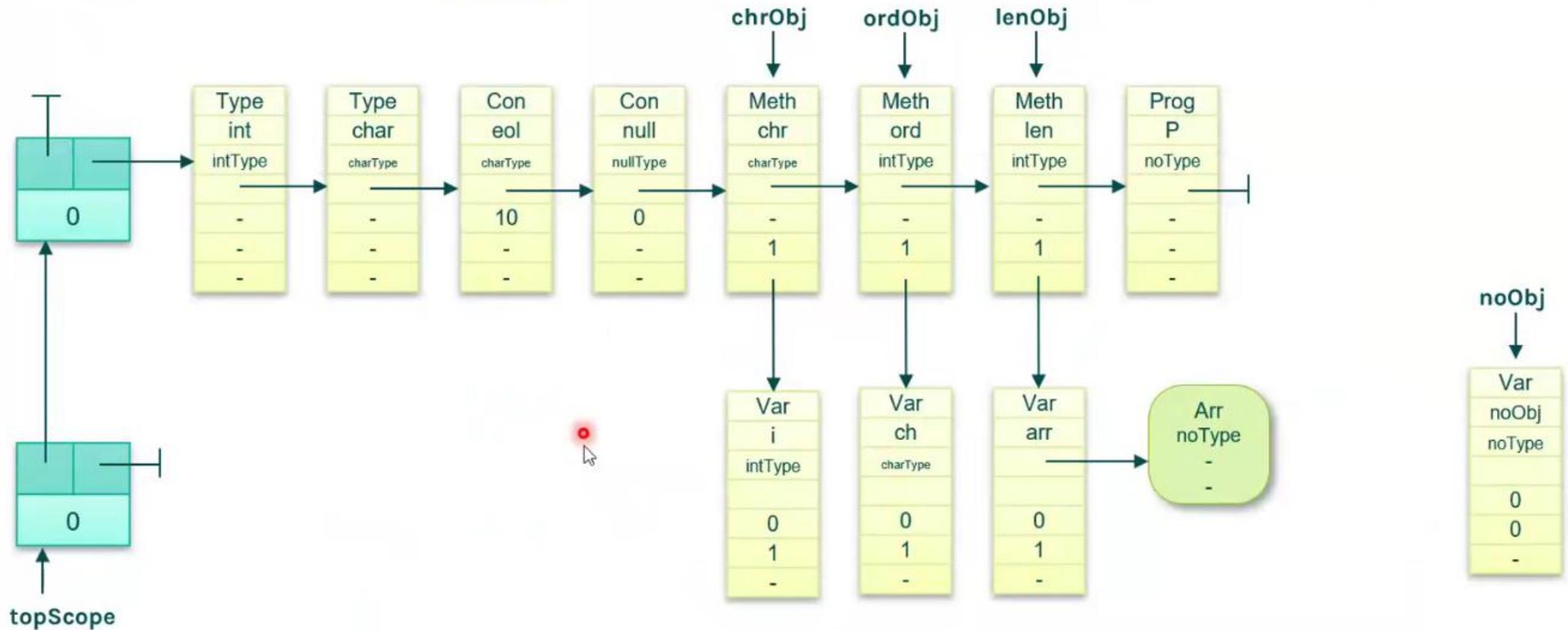
Broj deklariranih promenljivih unutar opsega.

Programski prevodioci 1

„Universe“ opseg



„Universe“ opseg



Programski prevodioci 1

Zadatak 1.

Prikazati izgled tabele simbola u karakterističnim tačkama (označenim sa (** x **)) za priloženi Mikrojava program.

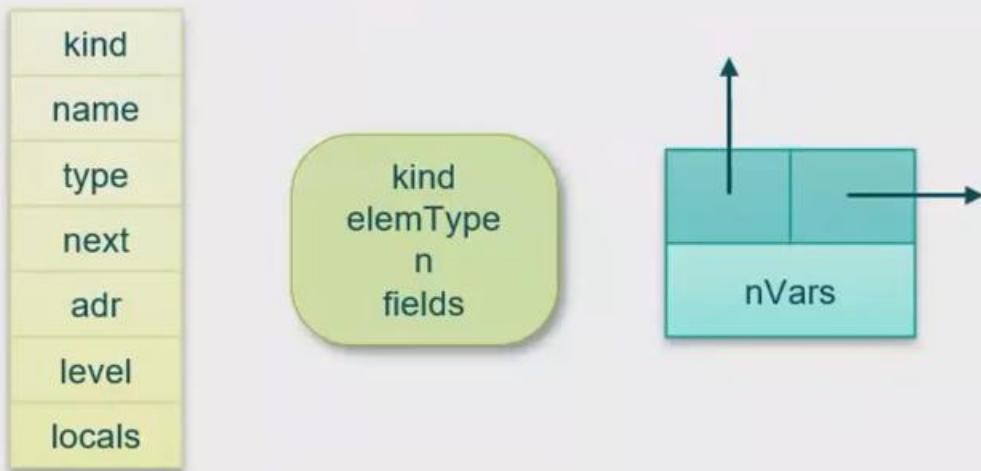
„Universe“ opseg nije potrebno crtati.

```
program ABC (** 1 **)
char c[];
int max;
char npp; (** 2 **)
{
    int put (** 3 **) (int x)
    { (** 4 **)
        return x;
    } (** 5 **)
} (** 6 **)
```

Zadatak 1.

Prikazati izgled tabele simbola u karakterističnim tačkama (označenim sa (** x **)) za priloženi Mikrojava program.

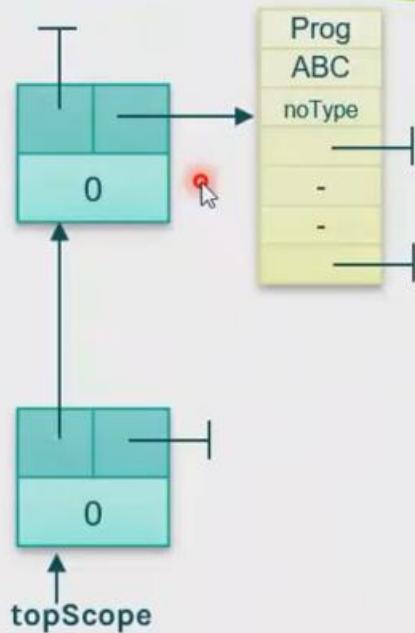
„Universe“ opseg nije potrebno crtati.



```
program ABC (** 1 **)
char c[];
int max;
char npp; (** 2 **)
{
    int put (** 3 **) (int x)
    { (** 4 **)
        return x;
    } (** 5 **)
} (** 6 **)
```

Programski prevodioci 1

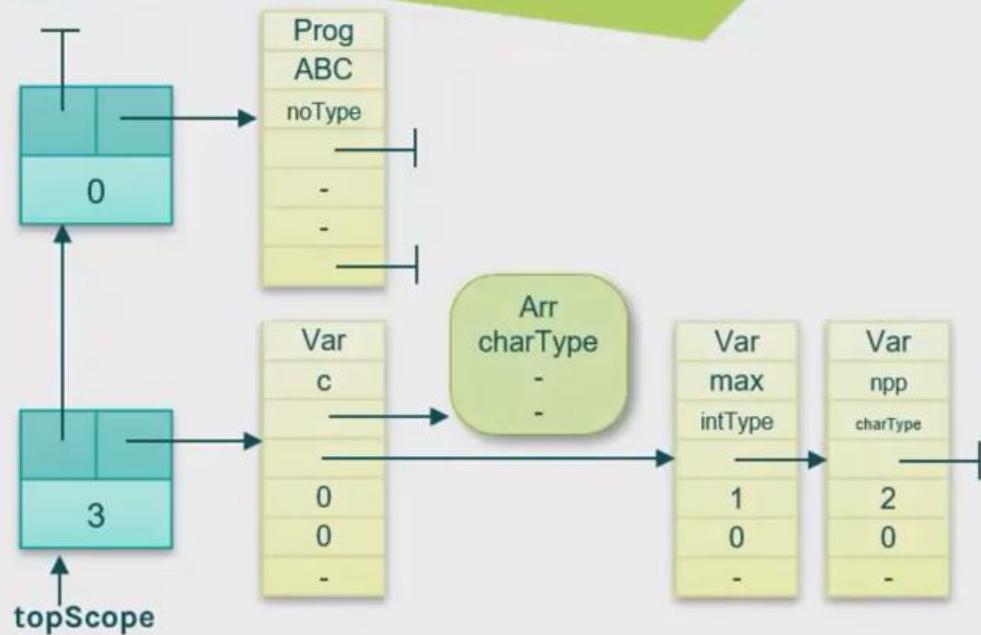
Zadatak 1.



```
program ABC (** 1 **)
char c[];
int max;
char npp; (** 2 **)
{
    int put (** 3 **) (int x)
    { (** 4 **)
        return x;
    } (** 5 **)
} (** 6 **)
```

Programski prevodioci 1

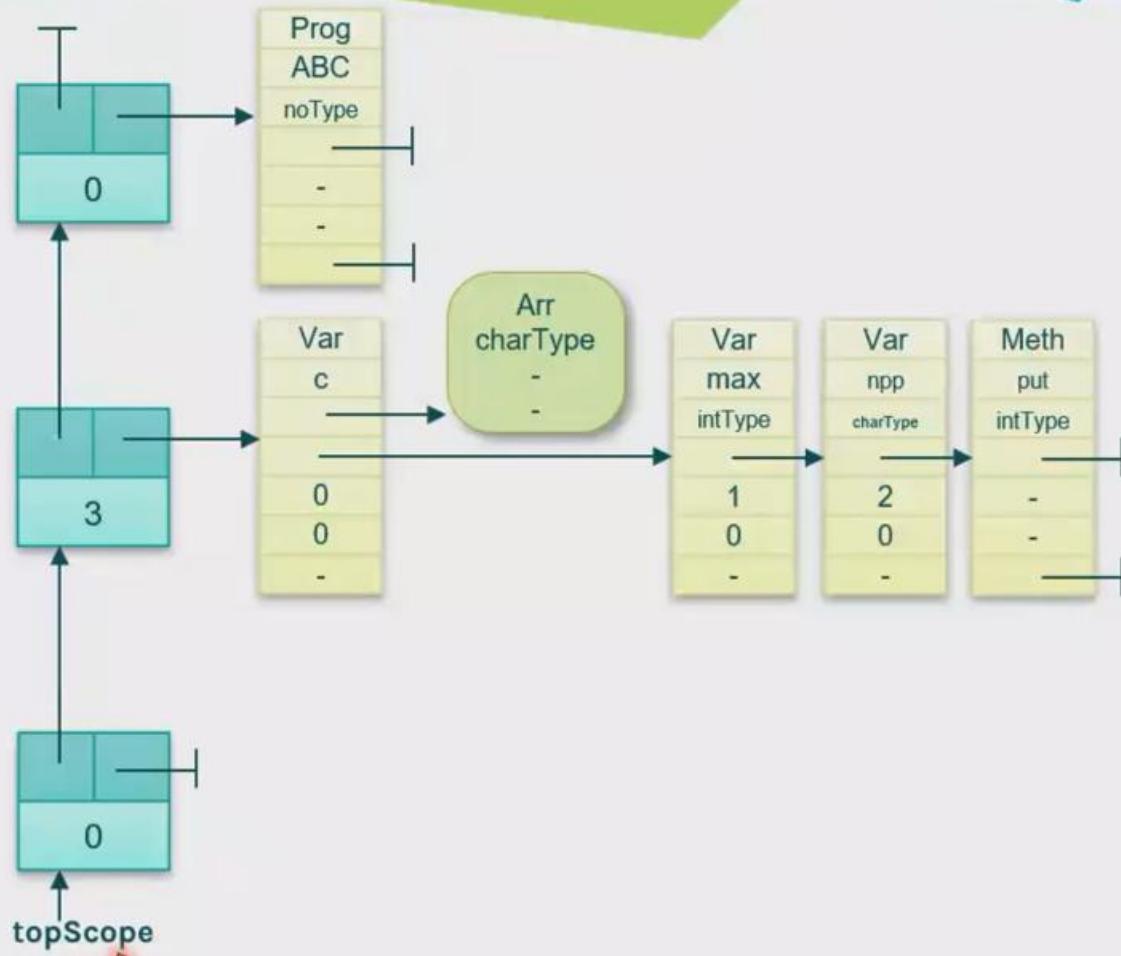
Zadatak 1.



```
program ABC (** 1 **)
char c[];
int max;
char npp; (** 2 **)
{
    int put (** 3 **) (int x)
    { (** 4 **)
        return x;
    } (** 5 **)
} (** 6 **)
```

Programski prevodioci 1

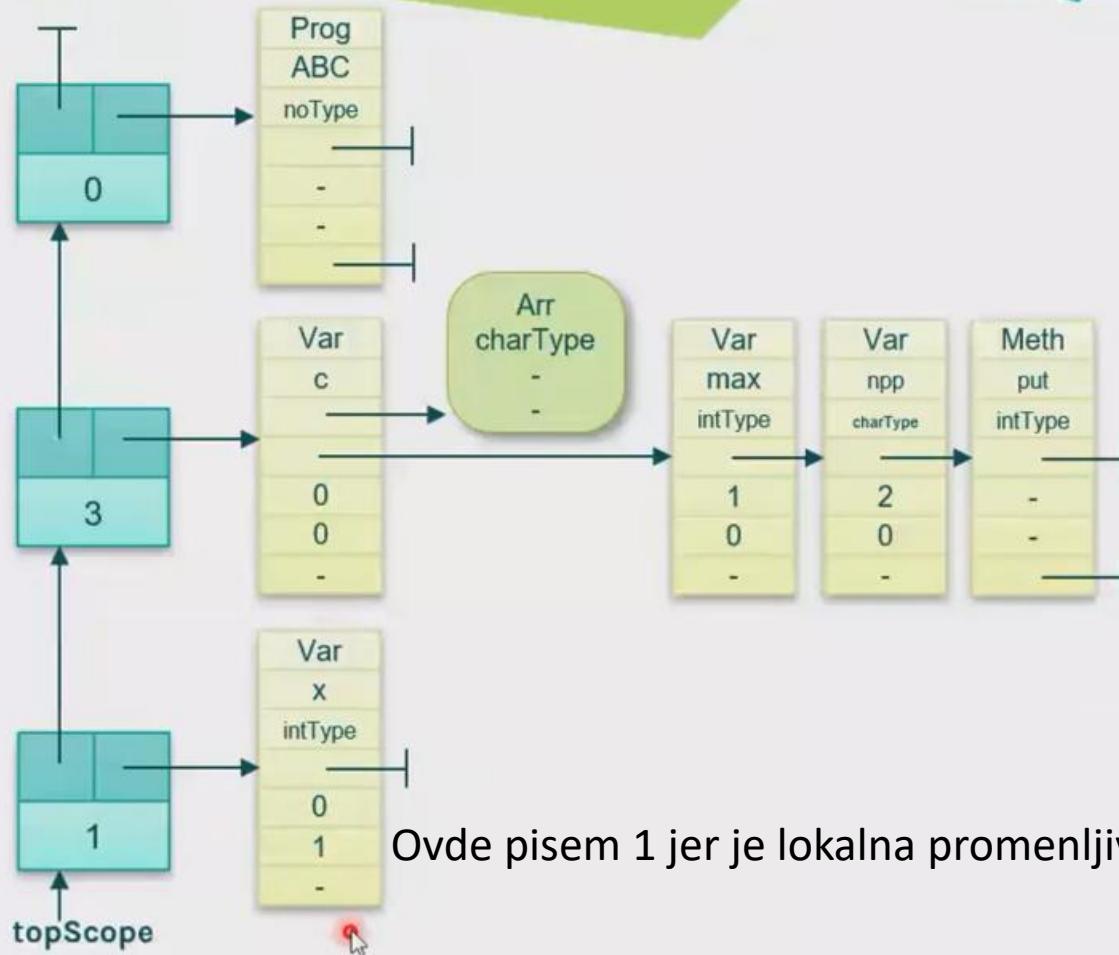
Zadatak 1.



Programski prevodioci 1

```
program ABC (** 1 **)
char c[];
int max;
char npp; (** 2 **)
{
    int put (** 3 **) (int x)
    { (** 4 **)
        return x;
    } (** 5 **)
} (** 6 **)
```

Zadatak 1.

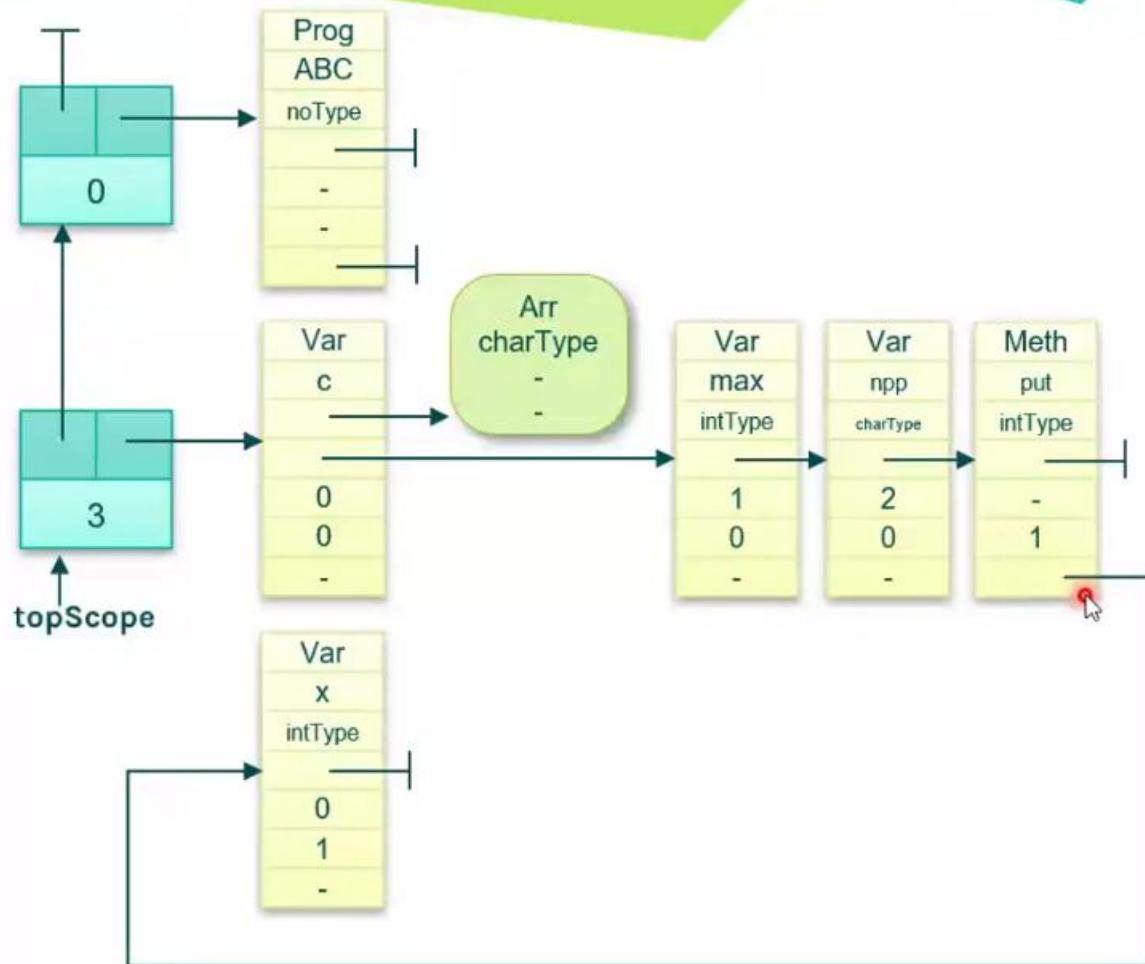


```
program ABC (** 1 **)
char c[];
int max;
char npp; (** 2 **)
{
    int put (** 3 **) (int x)
    { (** 4 **)
        return x;
    } (** 5 **)
} (** 6 **)
```

Ovde pisem 1 jer je lokalna promenljiva

Programski prevodioci 1

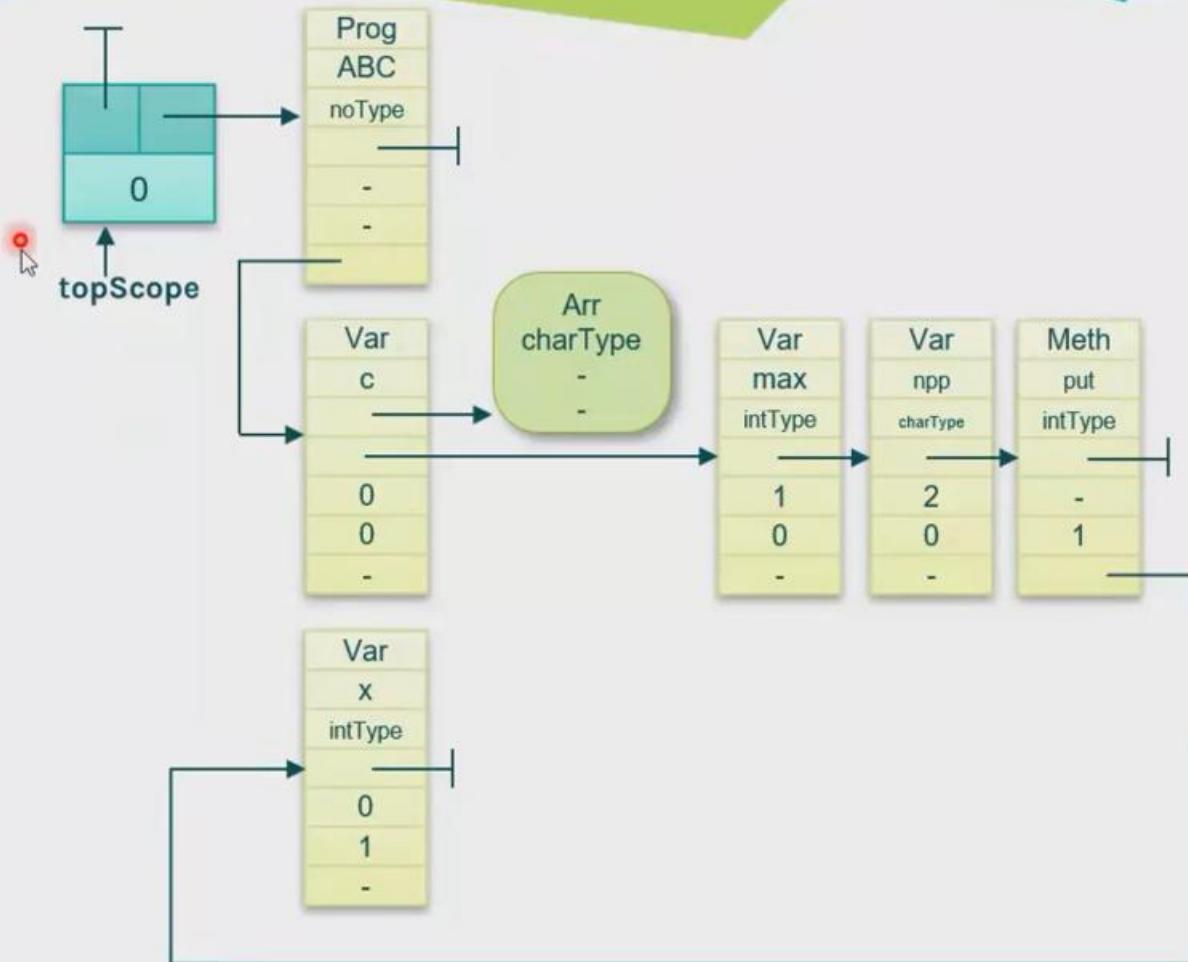
Zadatak 1.



```
program ABC (** 1 **)
char c[];
int max;
char npp; (** 2 **)
{
    int put (** 3 **) (int x)
    { (** 4 **)
        return x;
    } (** 5 **)
} (** 6 **)
```

Programski prevodioci 1

Zadatak 1.



Programski prevodioci 1

```
program ABC (** 1 **)
char c[];
int max;
char npp; (** 2 **)
{
    int put (** 3 **) (int x)
    { (** 4 **)
        return x;
    } (** 5 **)
} (** 6 **)
```

Zadatak 2.

Prikazati izgled tabele simbola u karakterističnim tačkama (označenim sa (** x **)) za priloženi Mikrojava program.

„Universe“ opseg nije potrebno crtati.

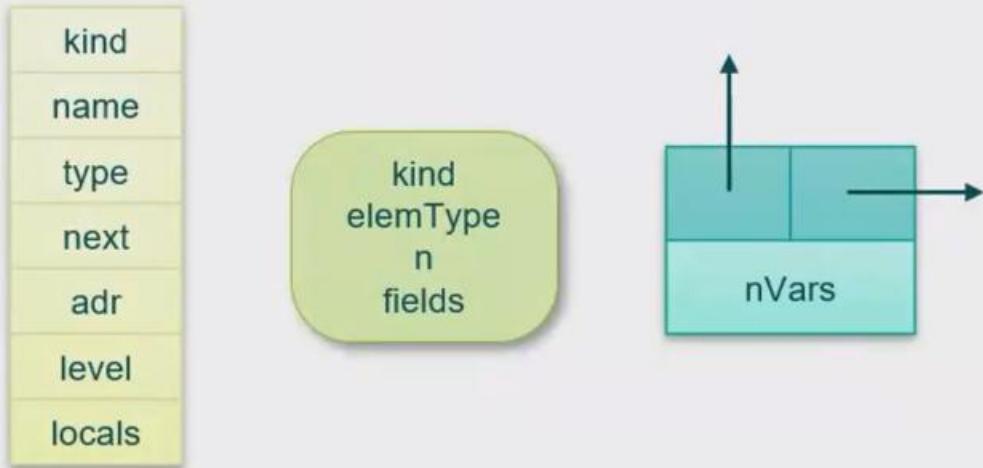
```
program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
void main () (** 8 **)
    int a, b; (** 9 **) {
        k = new Unutra;
        read(a); b = m1(a, k);
    } (** 10 **)
} (** 11 **)
```

Programski prevodioci 1

Zadatak 2.

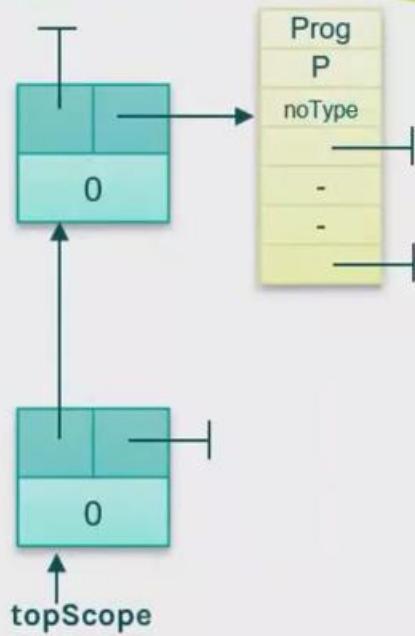
Prikazati izgled tabele simbola u karakterističnim tačkama (označenim sa (** x **)) za priloženi Mikrojava program.

„Universe“ opseg nije potrebno crtati.



```
program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
void main () (** 8 **)
    int a, b; (** 9 **) {
        k = new Unutra;
        read(a); b = m1(a, k);
    } (** 10 **)
} (** 11 **)
```

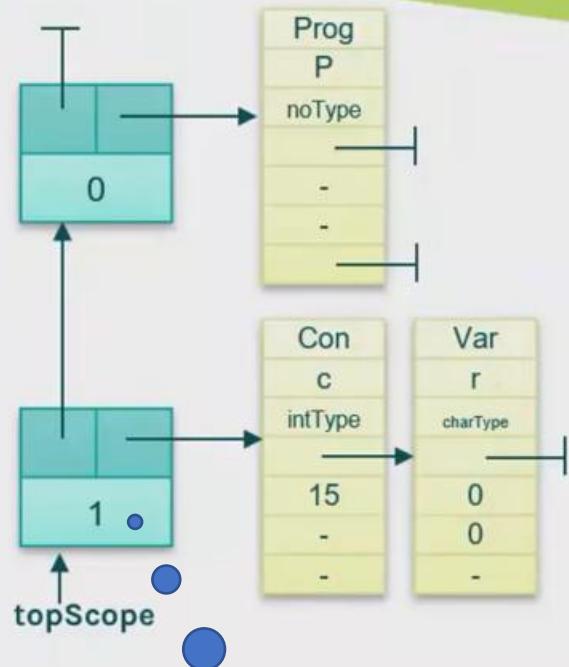
Zadatak 2.



```
program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
    void main () (** 8 **)
        int a, b; (** 9 **)
        k = new Unutra;
        read(a); b = m1(a, k);
    } (** 10 **)
} (** 11 **)
```

Programski prevodioci 1

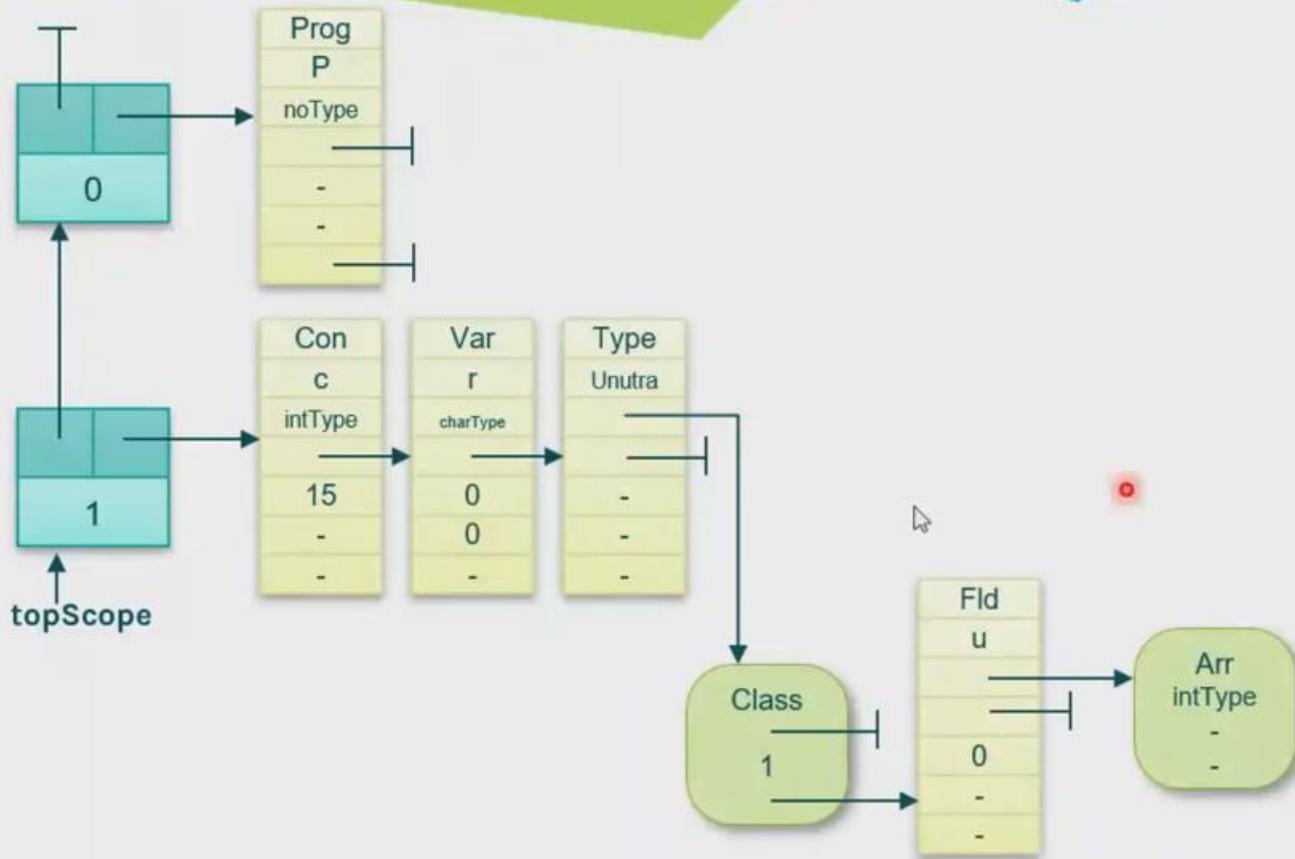
Zadatak 2.



Programski prevodioci 1

```
program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
    void main () (** 8 **)
        int a, b; (** 9 **)
        k = new Unutra;
        read(a); b = m1(a, k);
    } (** 10 **)
} (** 11 **)
```

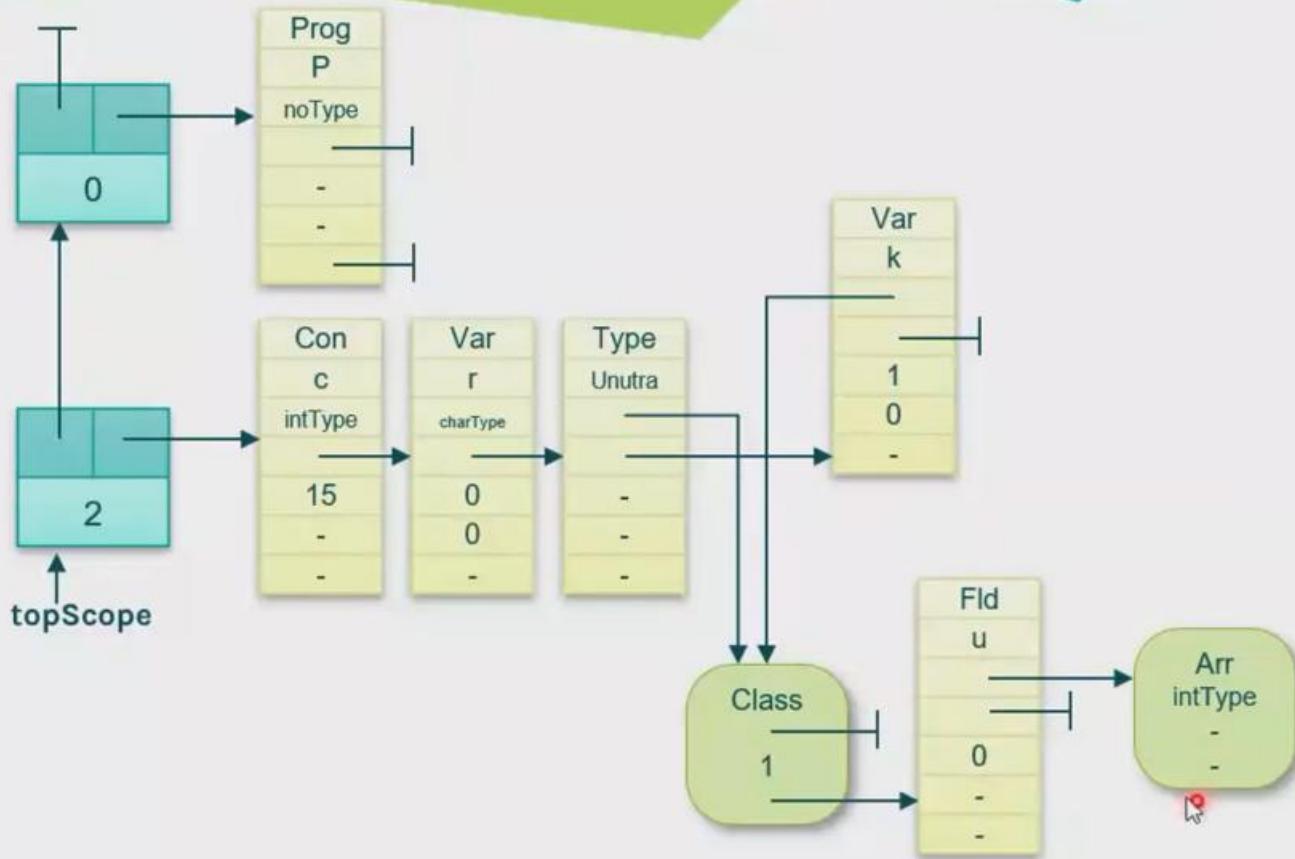
Zadatak 2.



```
program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
void main () (** 8 **)
    int a, b; (** 9 **)
    k = new Unutra;
    read(a); b = m1(a, k);
} (** 10 **)
} (** 11 **)
```

Programski prevodioci 1

Zadatak 2.

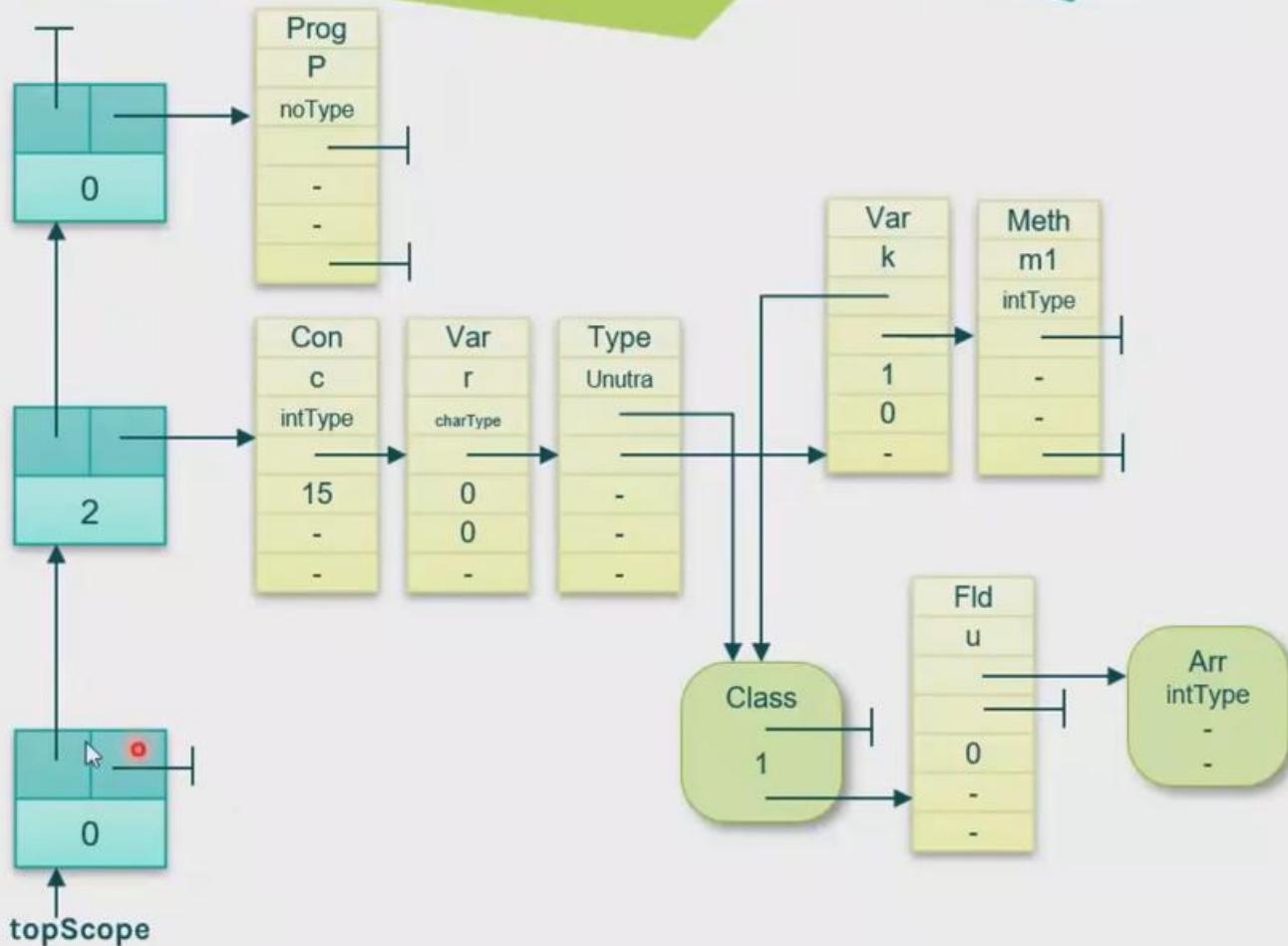


```
program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
void main () (** 8 **)
    int a, b; (** 9 **)
    k = new Unutra;
    read(a); b = m1(a, k);
} (** 10 **)
} (** 11 **)
```

Programski prevodioci 1

Zadatak 2.

```
program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
    void main () (** 8 **)
        int a, b; (** 9 **)
        k = new Unutra;
        read(a); b = m1(a, k);
    } (** 10 **)
} (** 11 **)
```



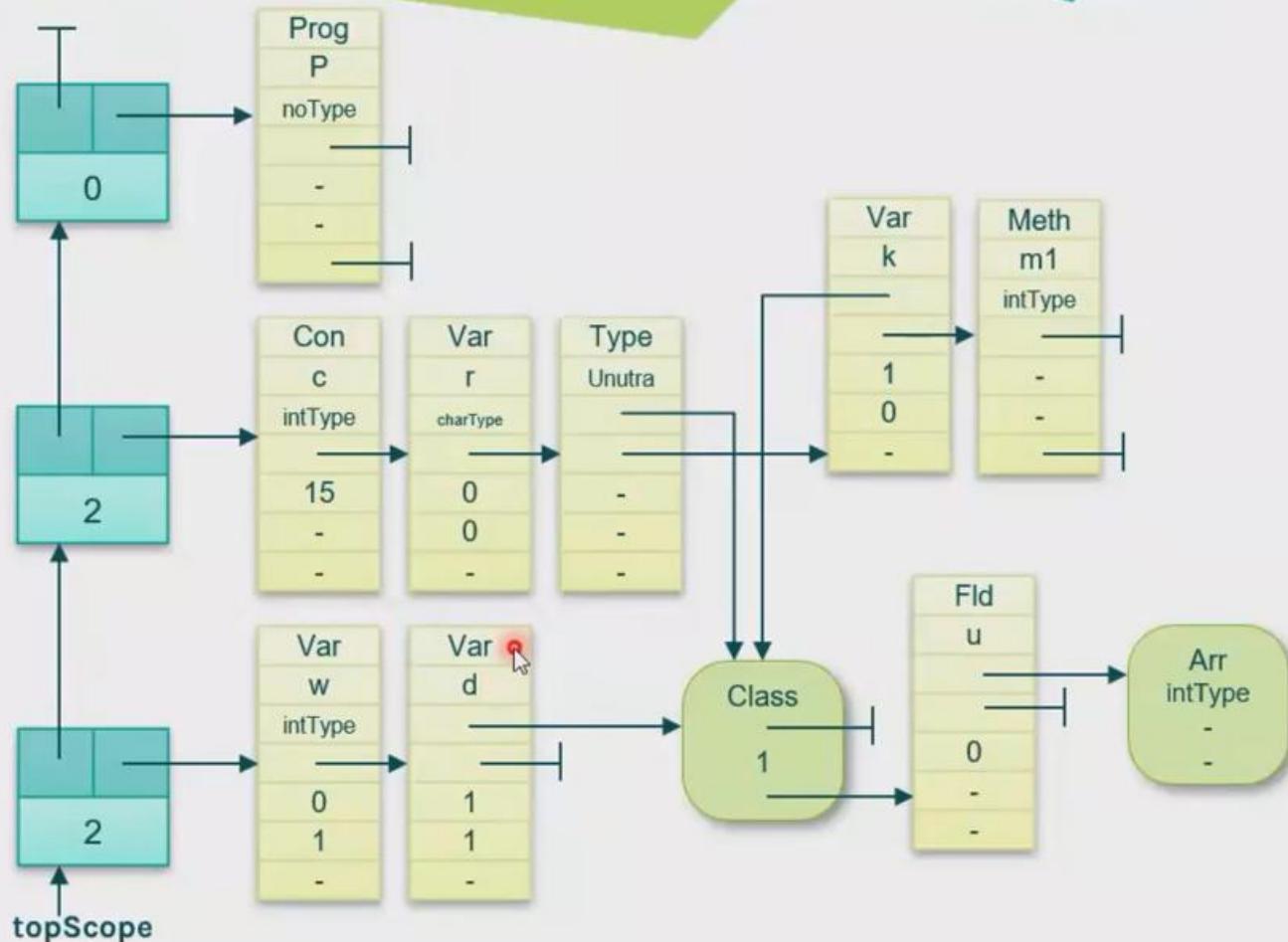
Programski prevodioci 1

Zadatak 2.

```

program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
void main () (** 8 **)
    int a, b; (** 9 **)
    k = new Unutra;
    read(a); b = m1(a, k);
} (** 10 **)
} (** 11 **)

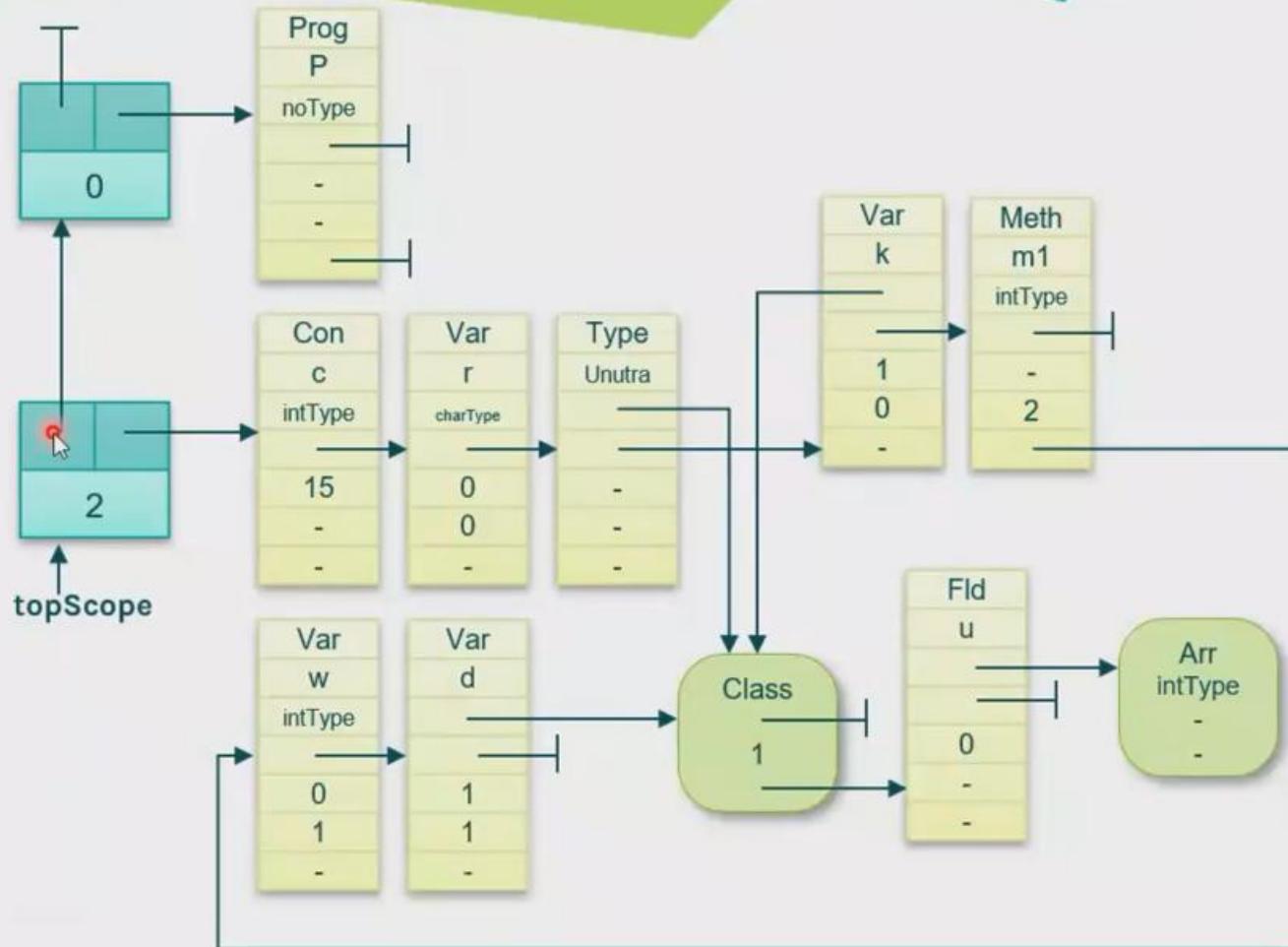
```



Programski prevodioci 1

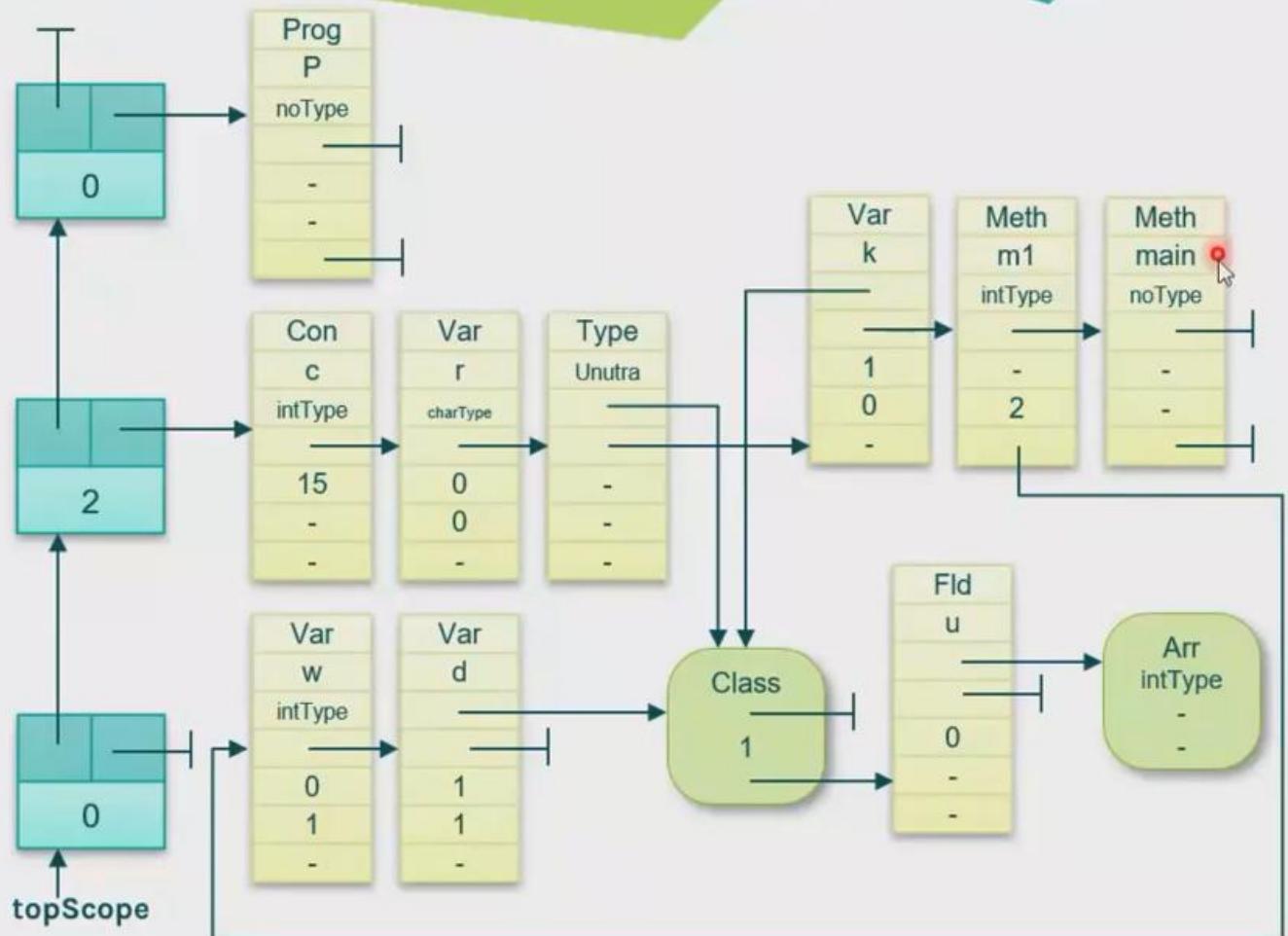
Zadatak 2.

```
program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
    void main () (** 8 **)
        int a, b; (** 9 **)
        k = new Unutra;
        read(a); b = m1(a, k);
    } (** 10 **)
} (** 11 **)
```



Programski prevodioci 1

Zadatak 2.

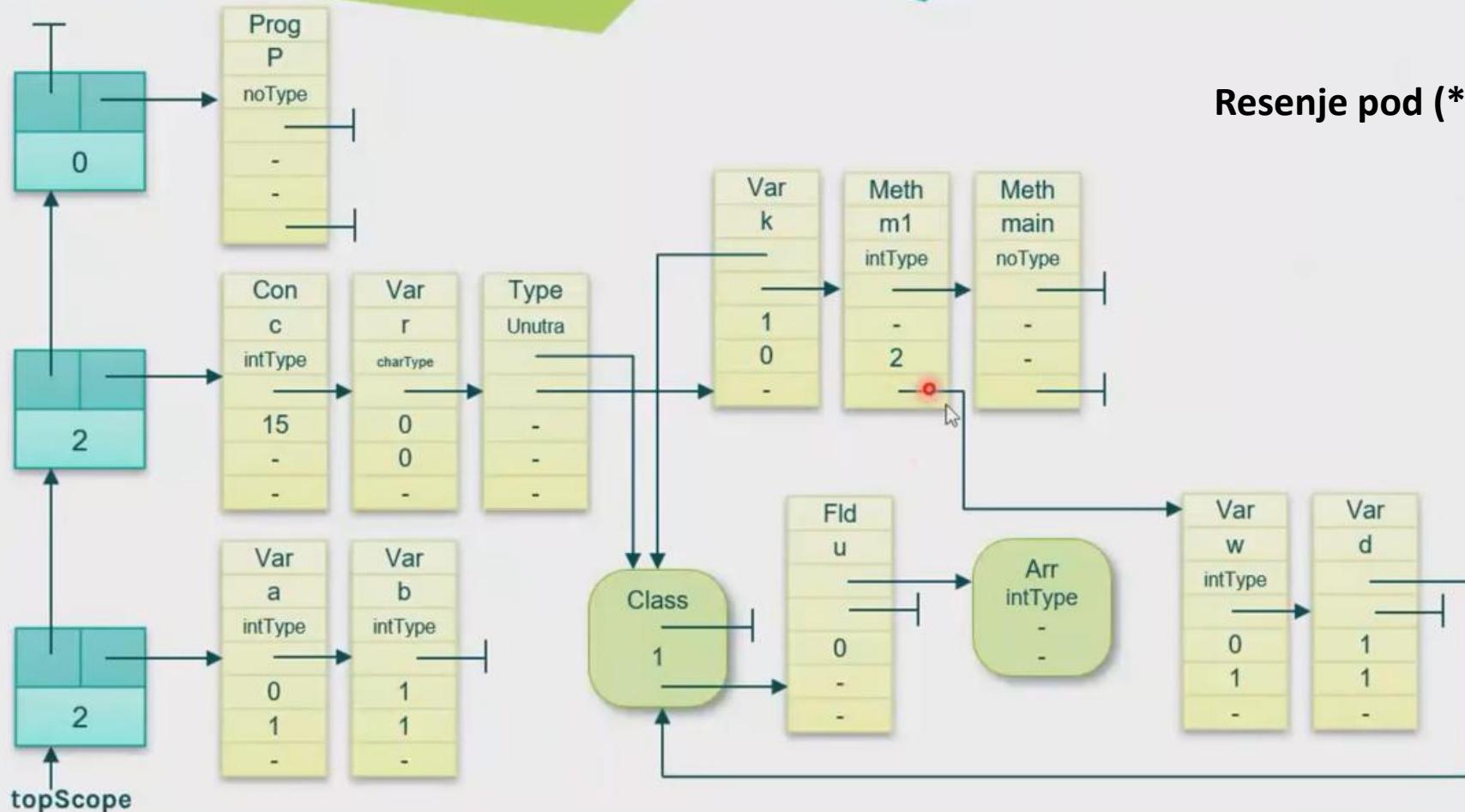


Programski prevodioci 1

Lokalne promenljive se nalaze pre {

Zadatak 2.

Resenje pod (** 9 **)



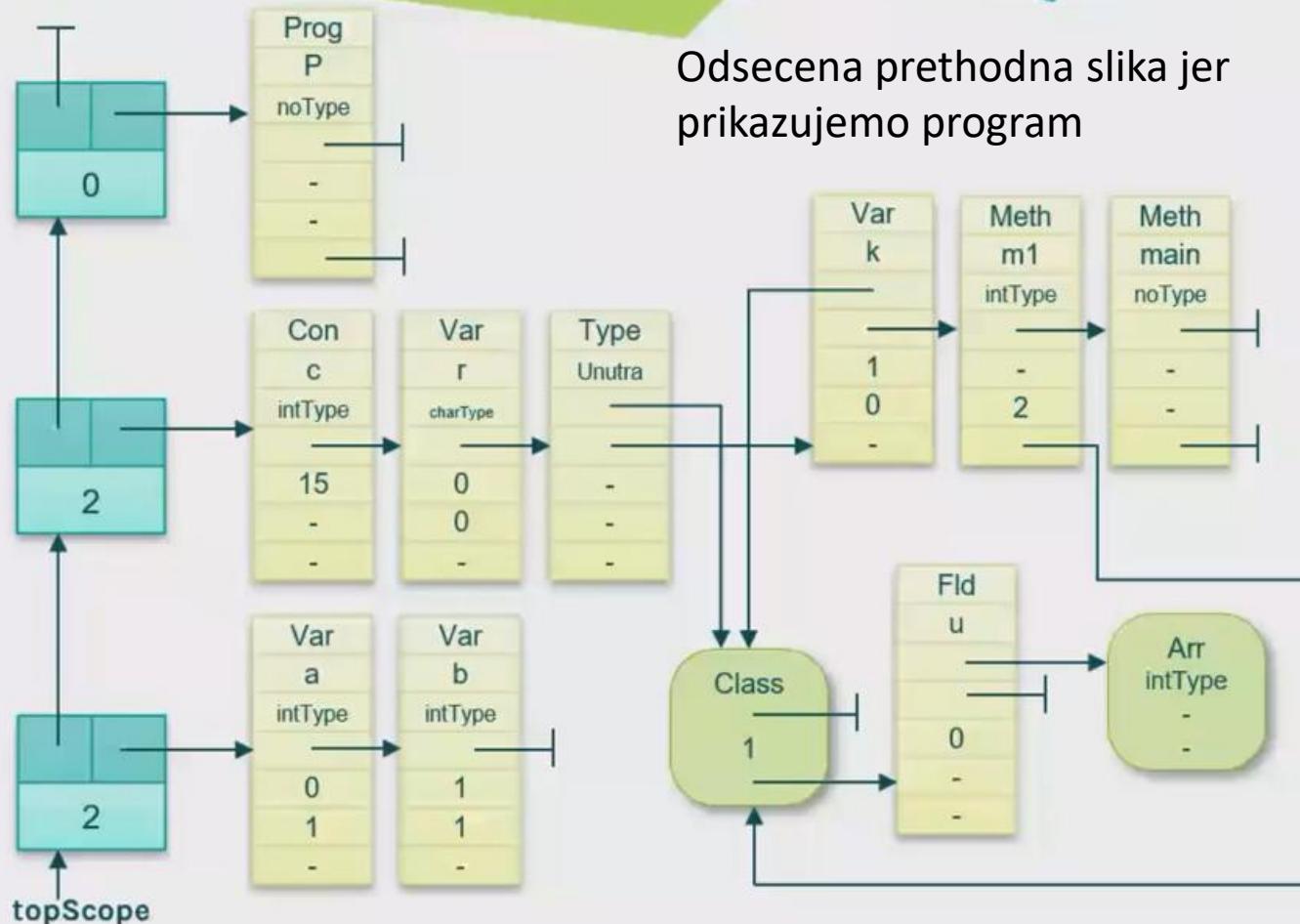
Programski prevodioci 1

Zadatak 2.

```

program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
    void main () (** 8 **)
        int a, b; (** 9 **)
        k = new Unutra;
        read(a); b = m1(a, k);
    } (** 10 **)
} (** 11 **)

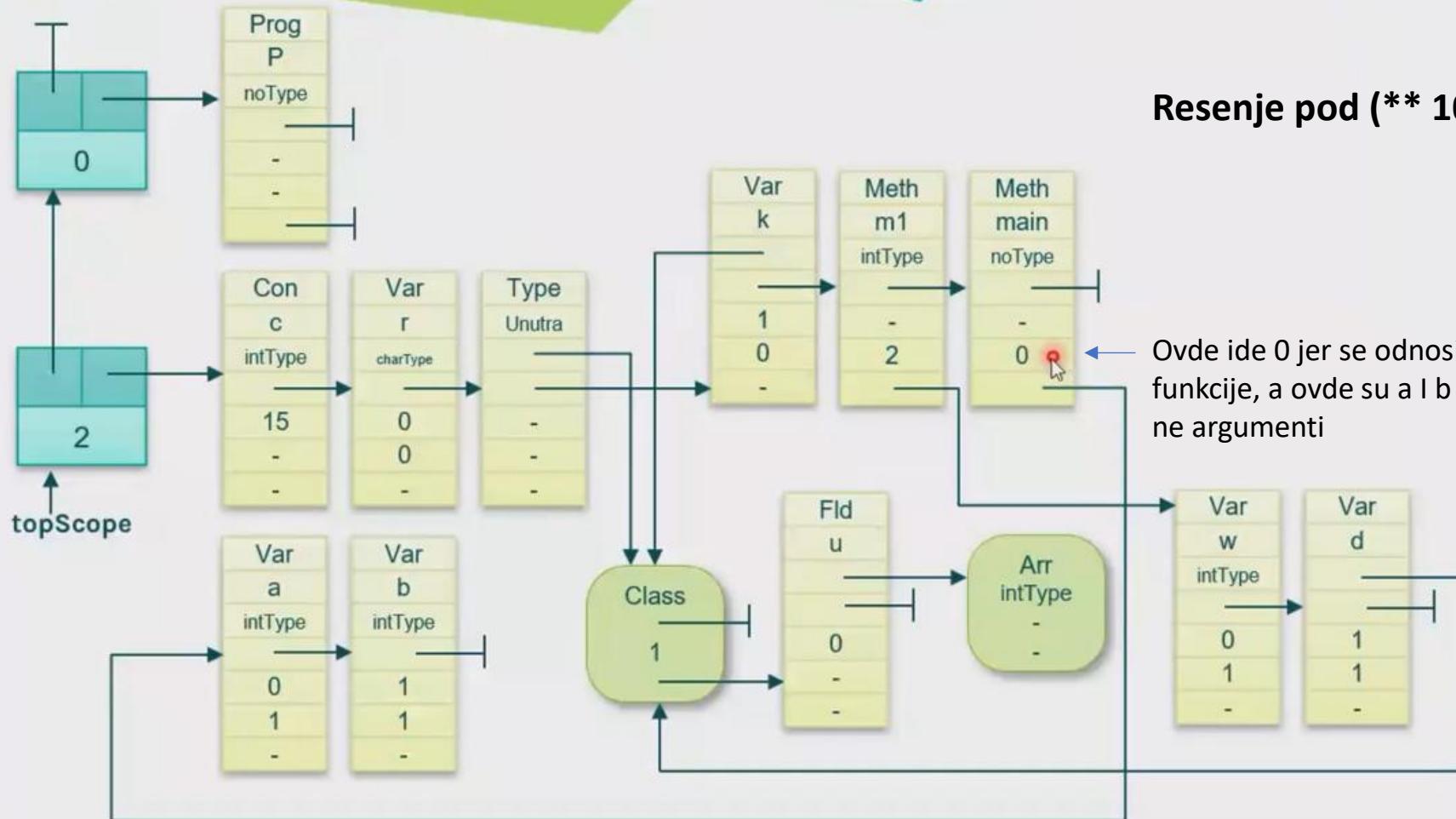
```



Programski prevodioci 1

Zadatak 2.

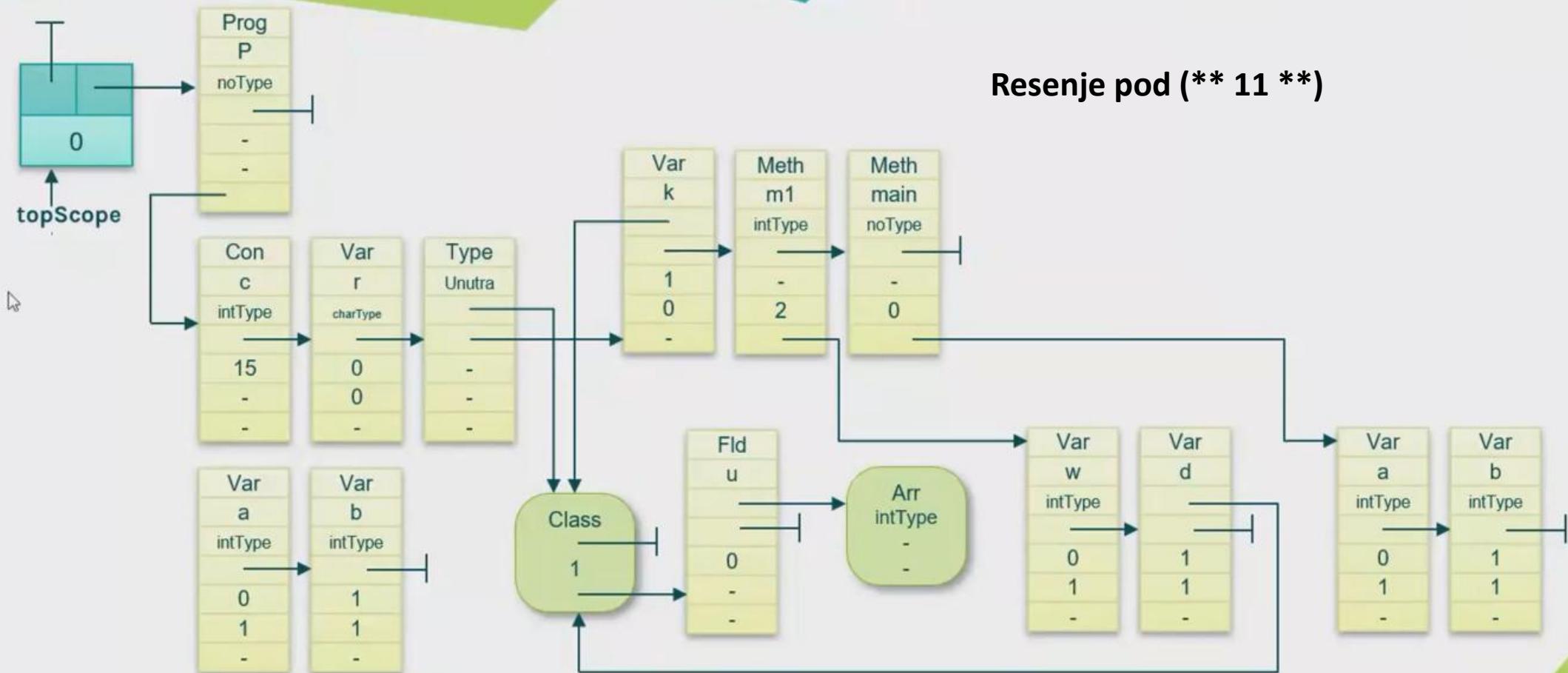
Resenje pod (** 10 **)



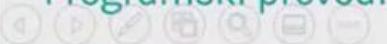
Programski prevodioci 1

Zadatak 2.

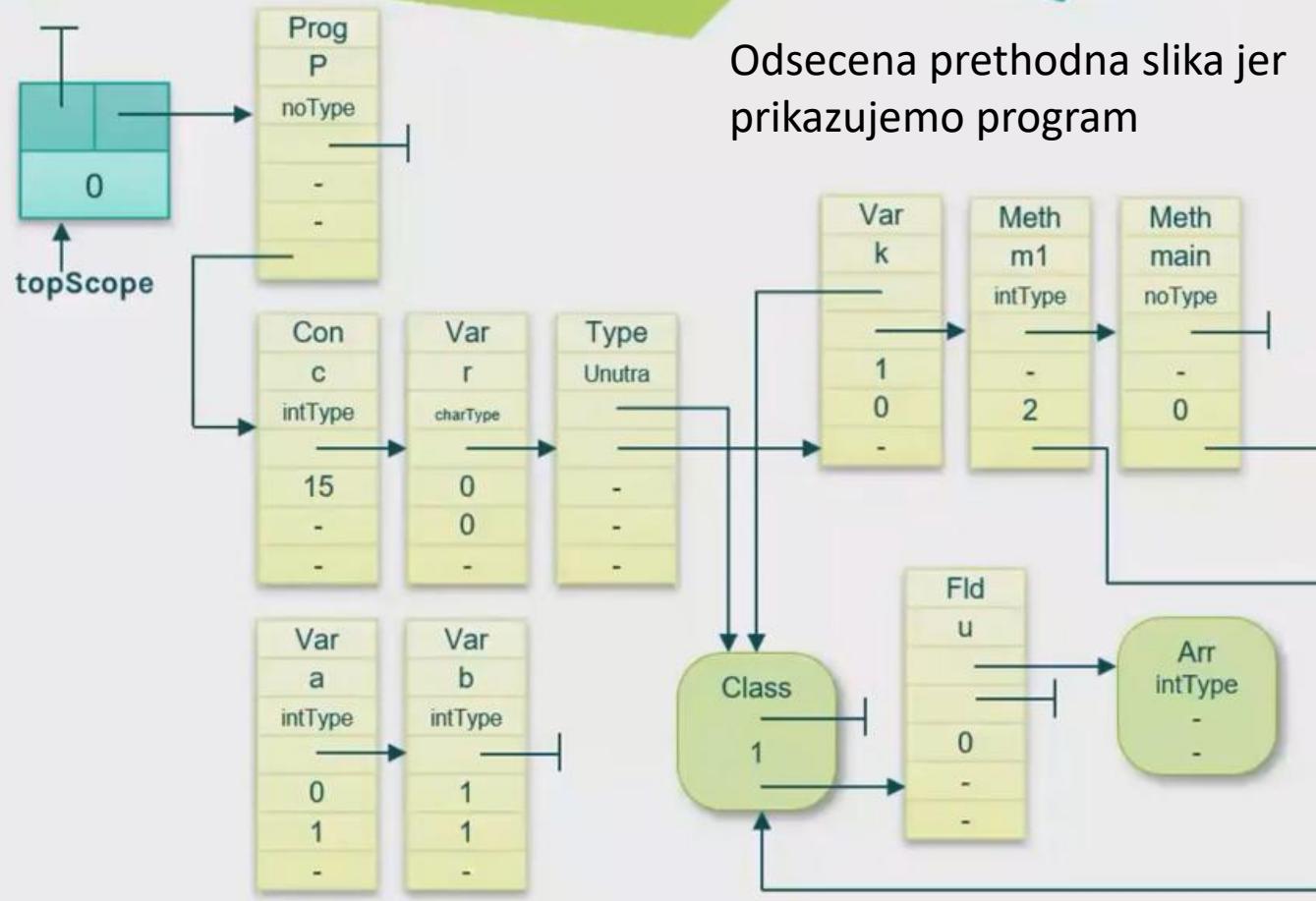
Resenje pod (** 11 **)



Programski prevodioci 1



Zadatak 2.



```

program P (** 1 **)
final int c = 15;
char r;
(** 2 **)
class Unutra{ int u[]; } (** 3 **)
Unutra k;
{ (** 4 **)
    int m1 (** 5 **)
        (int w, Unutra d)
    { (** 6 **)
        print(w);
        read(d.u[5]);
        return 0;
    } (** 7 **)
    void main () (** 8 **)
        int a, b; (** 9 **)
        k = new Unutra;
        read(a); b = m1(a, k);
    } (** 10 **)
} (** 11 **)

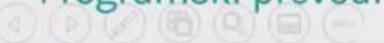
```

Zadatak 3.

Prikazati izgled tabele simbola u karakterističnim tačkama (označenim sa (** x **)) za priloženi Mikrojava program.

„Universe“ opseg nije potrebno crtati.

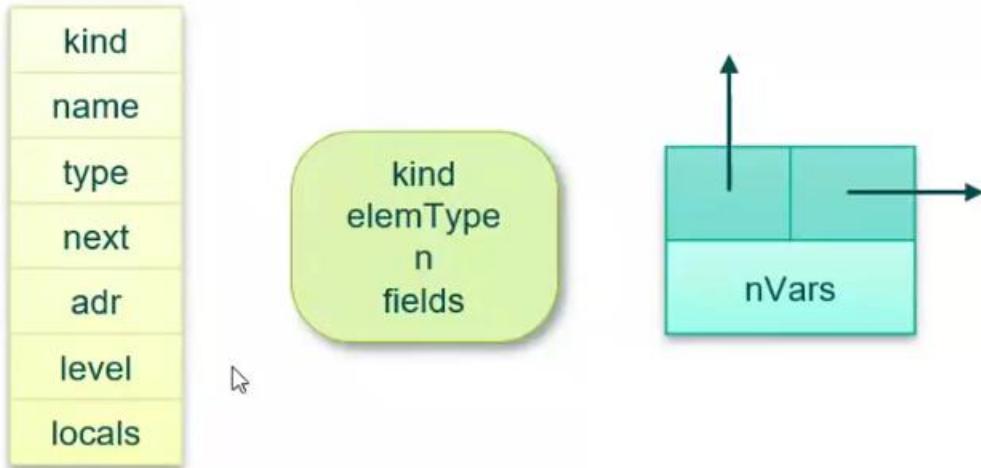
```
program OOMJ (** 1 **)
class TCalc (* T2 *) {
    double tax; (* T3 *)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (* T4 *)
} (* T5 *)
class ATCalc (* T6 *) extends TCalc {
    double a;
{ (* T7 *)
    double calc(double c) { return ... }
} (* T8 *)
} (* T9 *)
{
    void main()
        TCalc c1, c2; double cost; (* T10 *)
{
    ...
}
} (* T11 *)
```



Zadatak 3.

Prikazati izgled tabele simbola u karakterističnim tačkama (označenim sa (** x **)) za priloženi Mikrojava program.

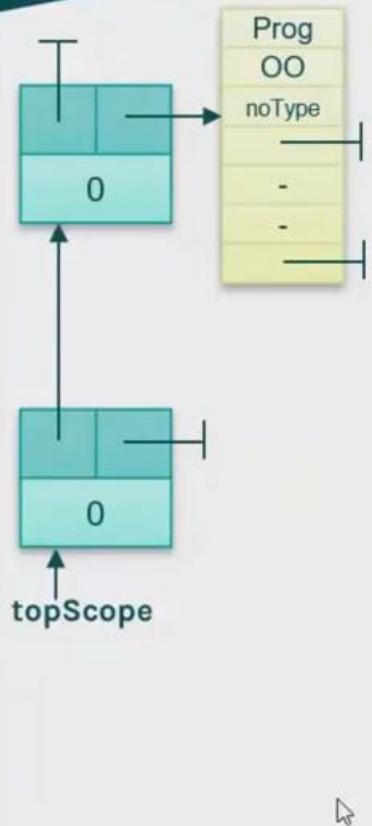
„Universe“ opseg nije potrebno crtati.



Programski prevodioci 1

```
program OOMJ (** 1 **)
class TCalc (* T2 *) {
    double tax; (* T3 *)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (* T4 *)
} (* T5 *)
class ATCalc (* T6 *) extends TCalc {
    double a;
{ (* T7 *)
    double calc(double c) { return ... }
} (* T8 *)
} (* T9 *)
{
    void main()
        TCalc c1, c2; double cost; (* T10 *)
{
    ...
} (* T11 *)
```

Zadatak 3.

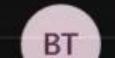


```
program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
{ (** T7 **)
    double calc(double c) { return ... }
} (** T8 **)
} (** T9 **)
{
    void main()
        TCalc c1, c2; double cost; (** T10 **)
    {
        ...
    }
} (** T11 **)
```

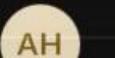
Programski prevodioci 1



+7



BT



AH



МК



УС



KZ



СЧ



АК



KM



PB

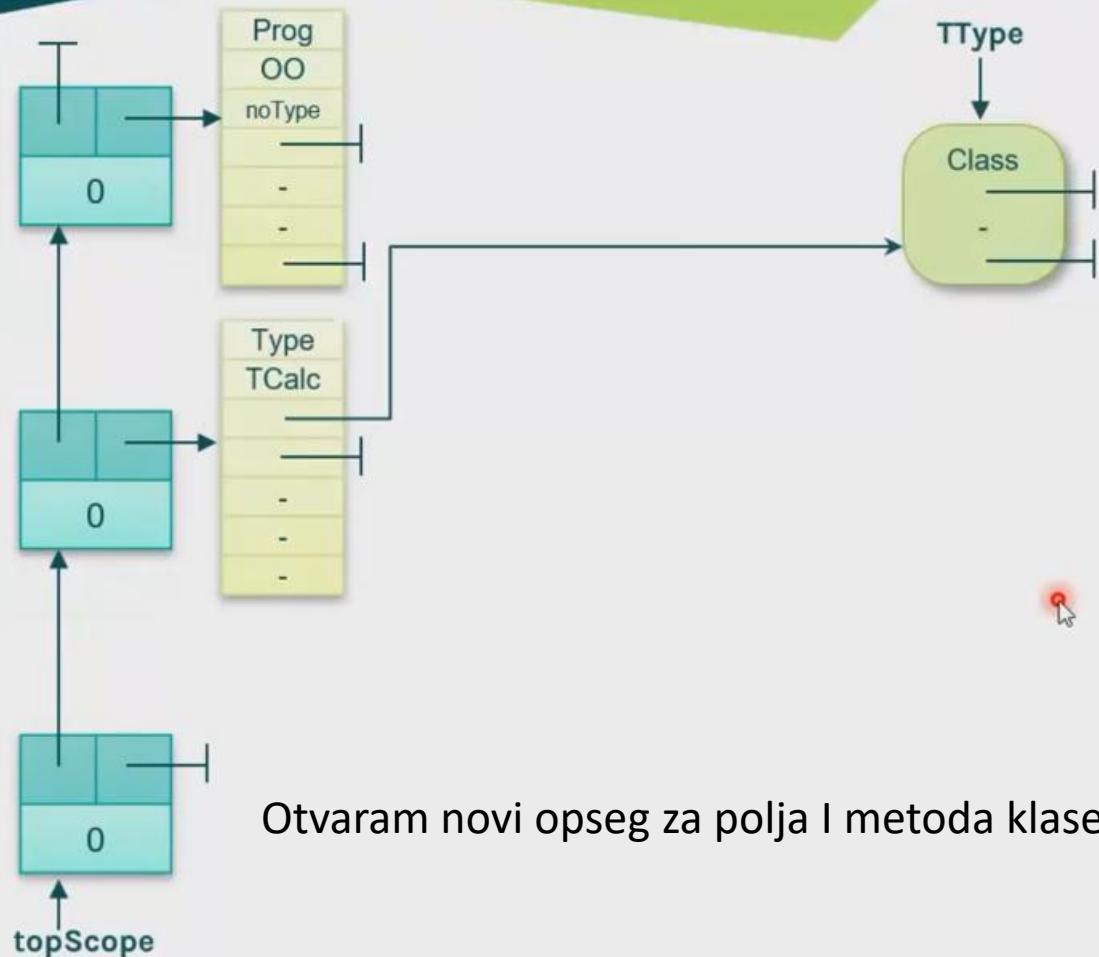


НБ



ФС

Zadatak 3.

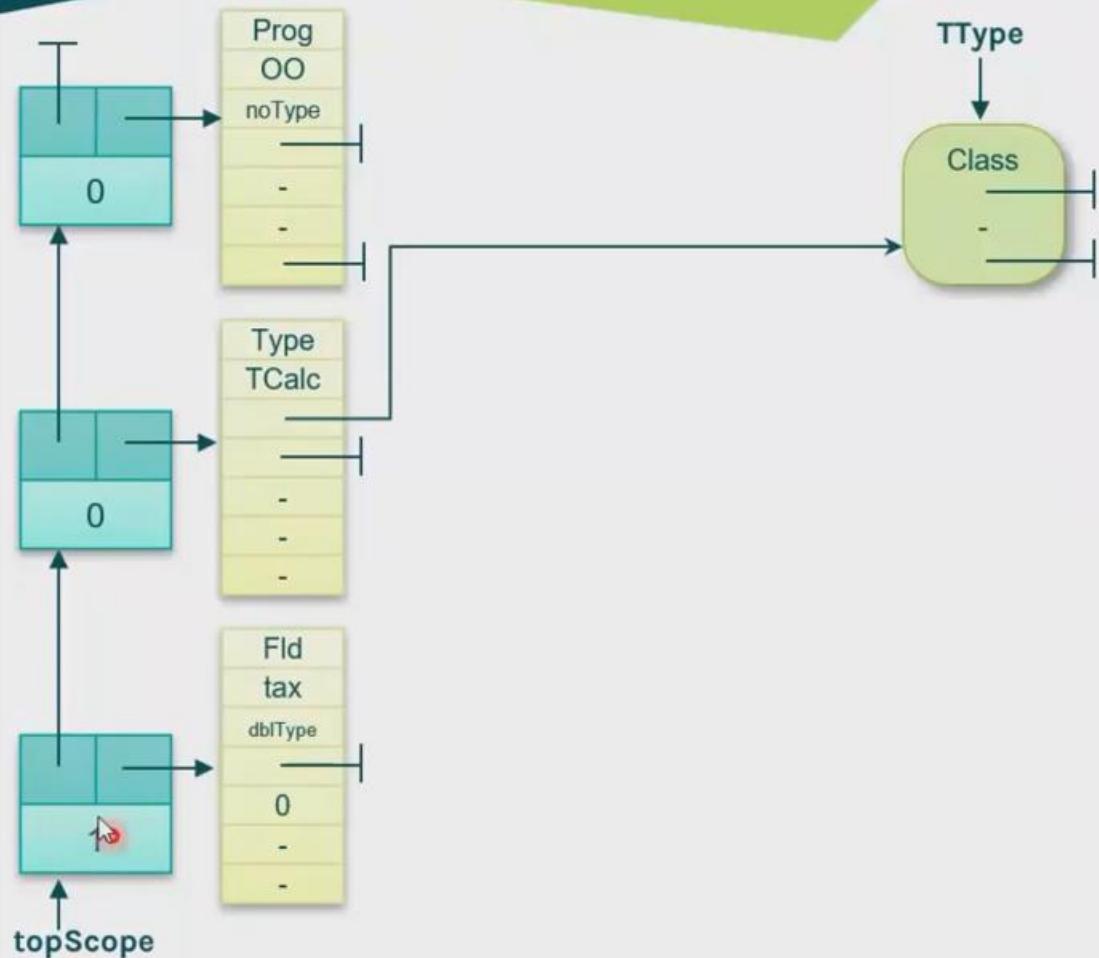


Otvaram novi opseg za polja i metoda klase

```
program OO (** 1 **)
class TCalc (** T2 **)
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
    { (** T7 **)
        double calc(double c) { return ... }
    } (** T8 **)
} (** T9 **)
{
    void main()
        TCalc c1, c2; double cost; (** T10 **)
    {
        ...
    } (** T11 **)
```

Programski prevodioci 1

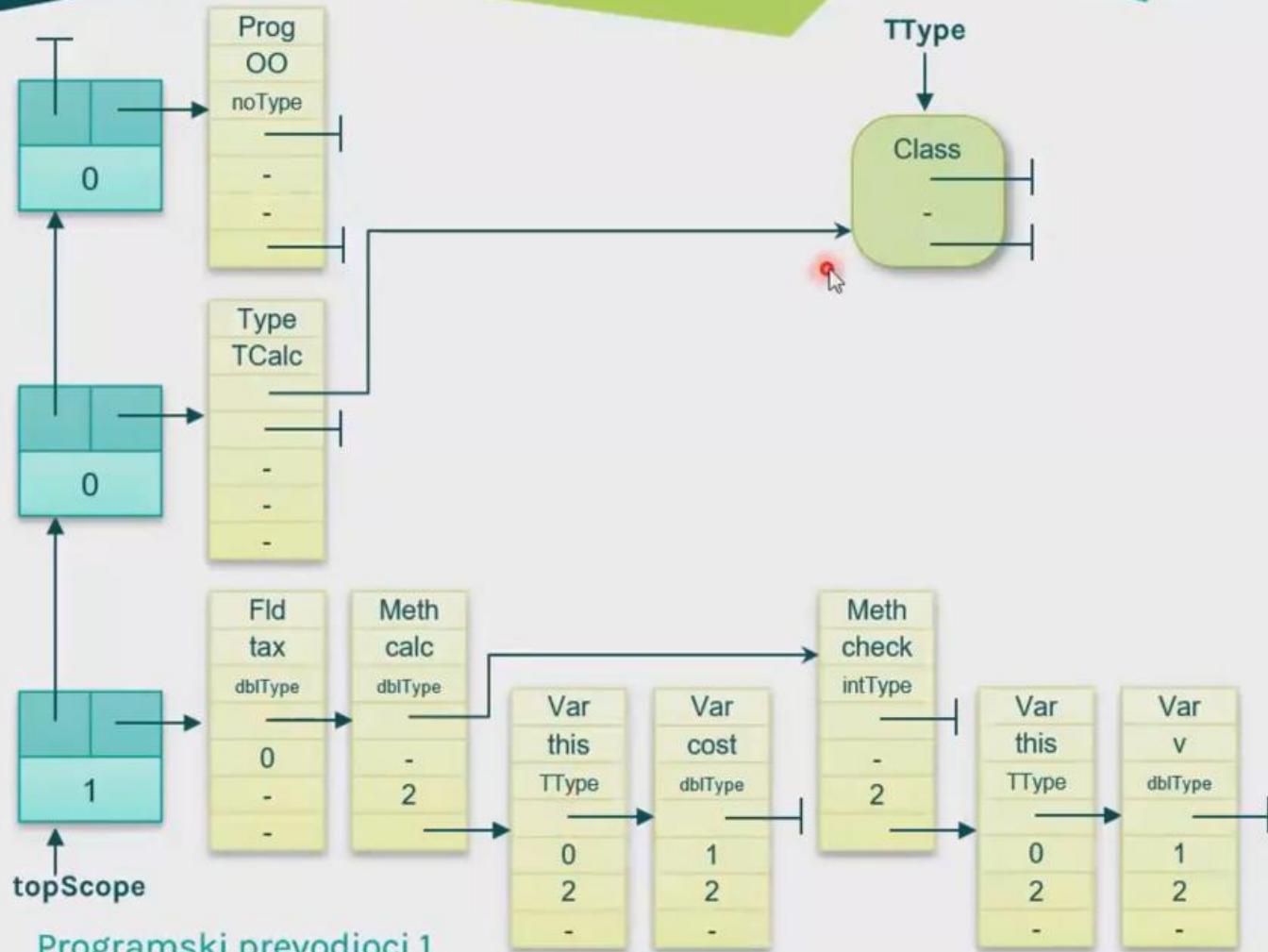
Zadatak 3.



Programski prevodioci 1

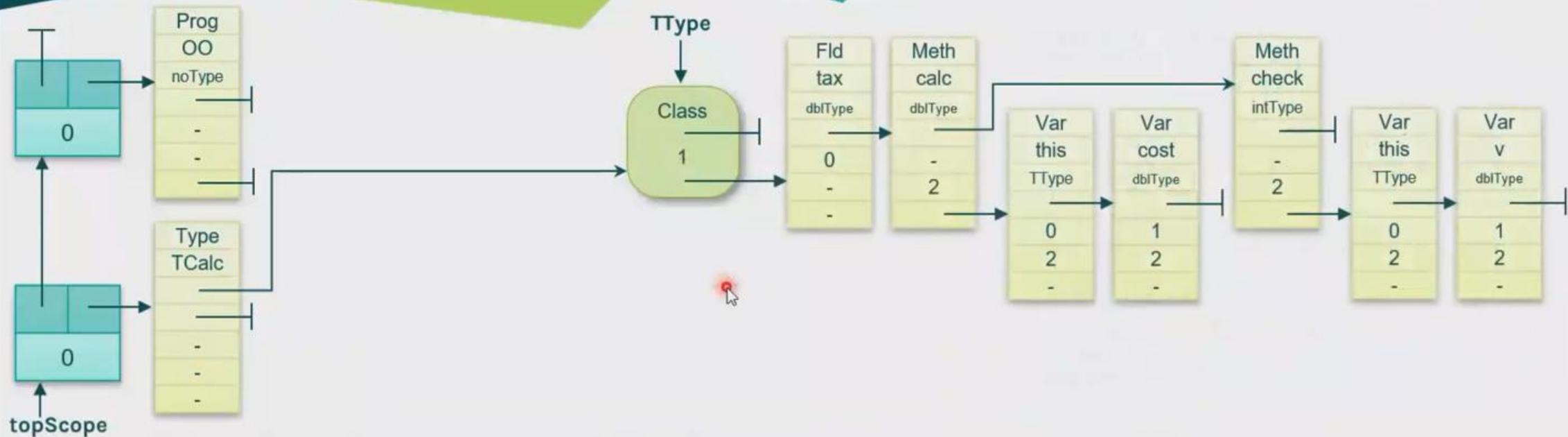
```
program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
{ (** T7 **)
    double calc(double c) { return ... }
} (** T8 **)
} (** T9 **)
{
    void main()
        TCalc c1, c2; double cost;(** T10 **)
    {
        ...
    } (** T11 **)
```

Zadatak 3.



```
program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
{ (** T7 **)
    double calc(double c) { return ... }
} (** T8 **)
} (** T9 **)
{
void main()
    TCalc c1, c2; double cost; (** T10 **)
{
    ...
} (** T11 **)
```

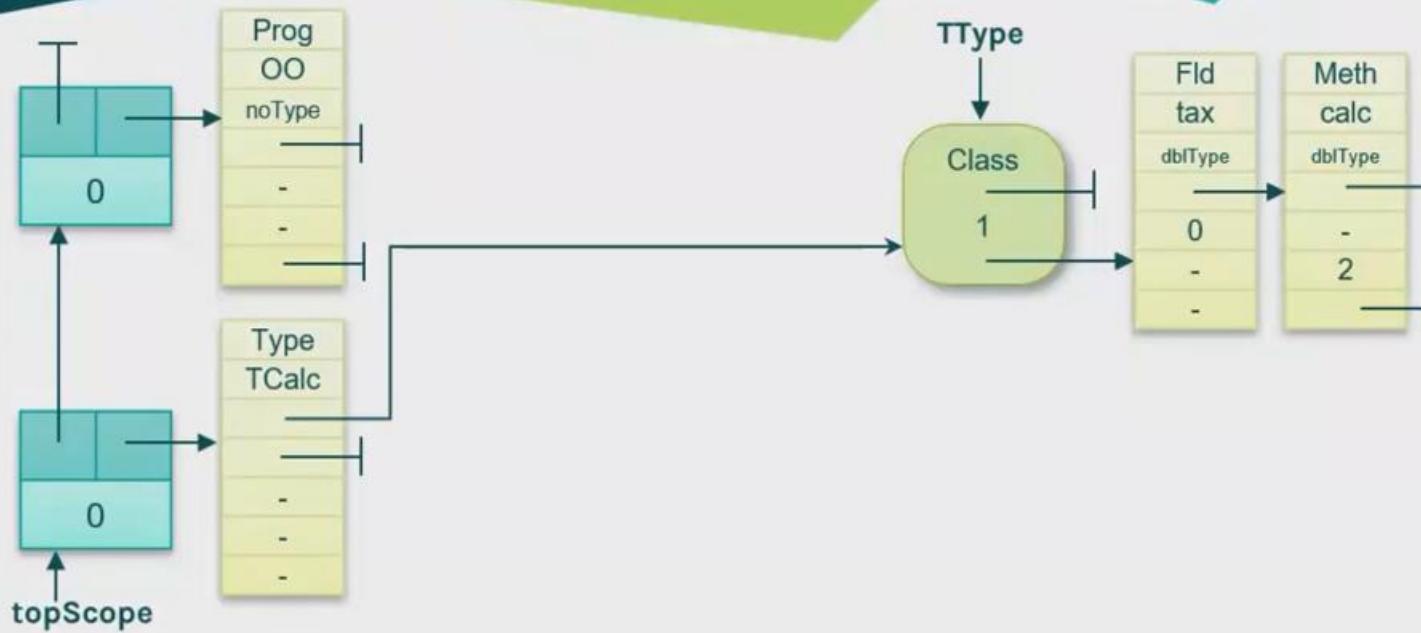
Zadatak 3.



Resenje (** T5 **)

Programski prevodioci 1

Zadatak 3.

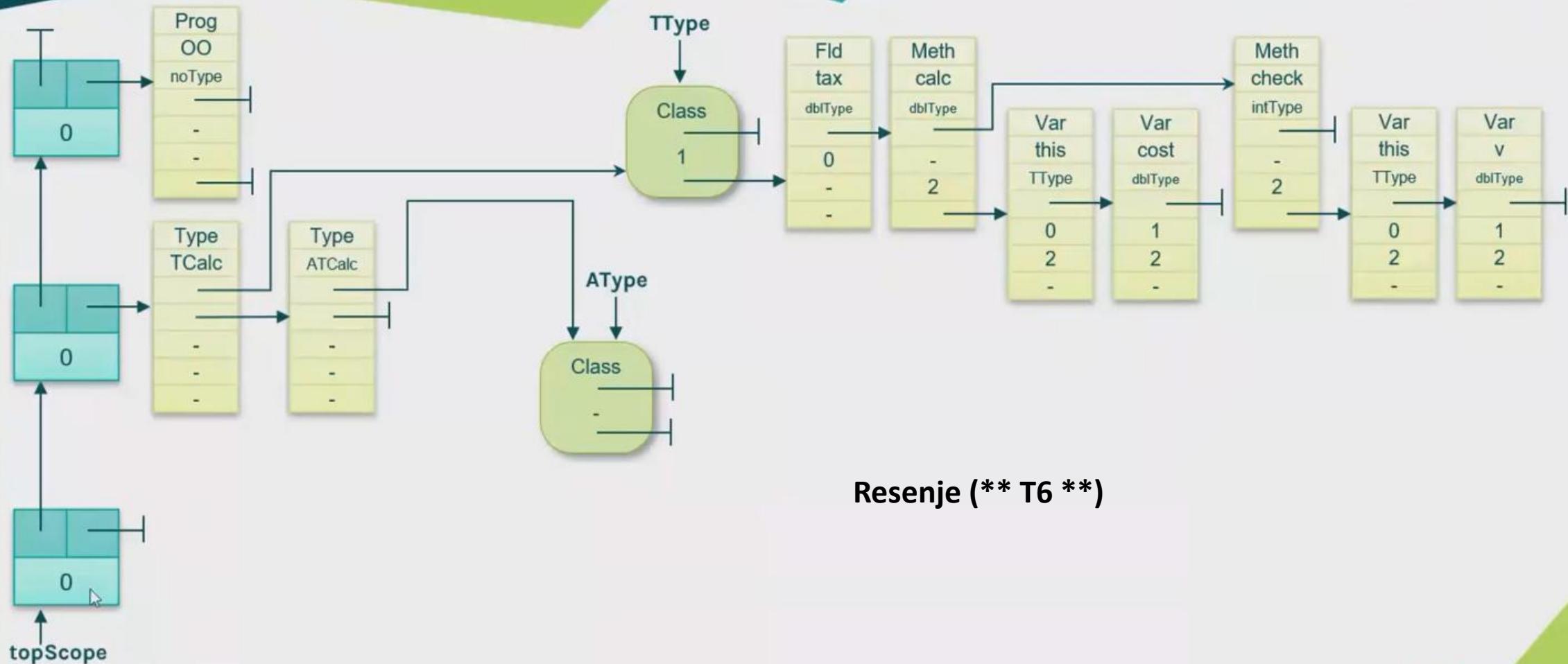


```

program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
{ (** T7 **)
    double calc(double c) { return ... }
} (** T8 **)
} (** T9 **)
{
void main()
    TCalc c1, c2; double cost;(** T10 **)
{
    ...
}
} (** T11 **)

```

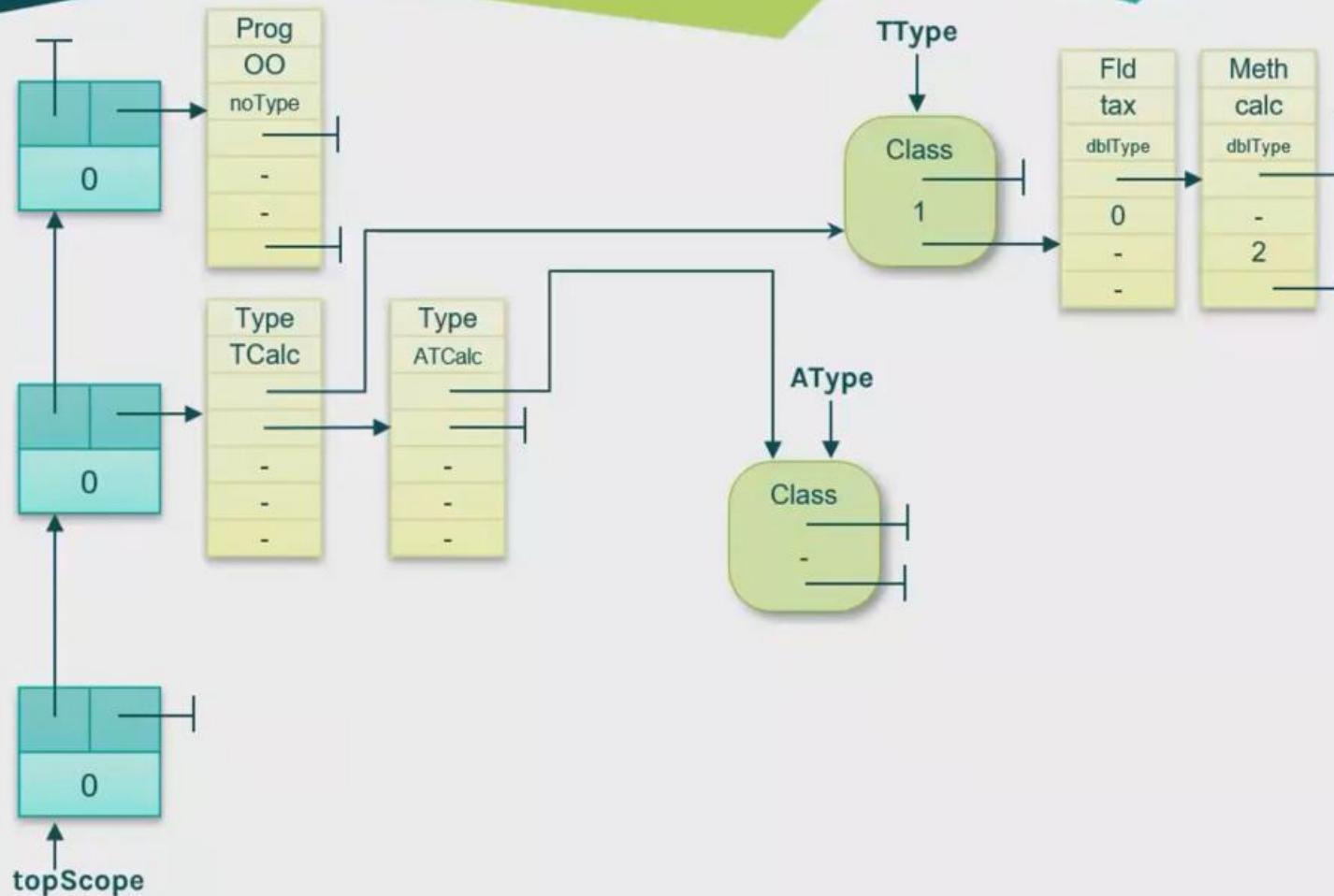
Zadatak 3.



Resenje (** T6 **)

Programski prevodioci 1

Zadatak 3.



```

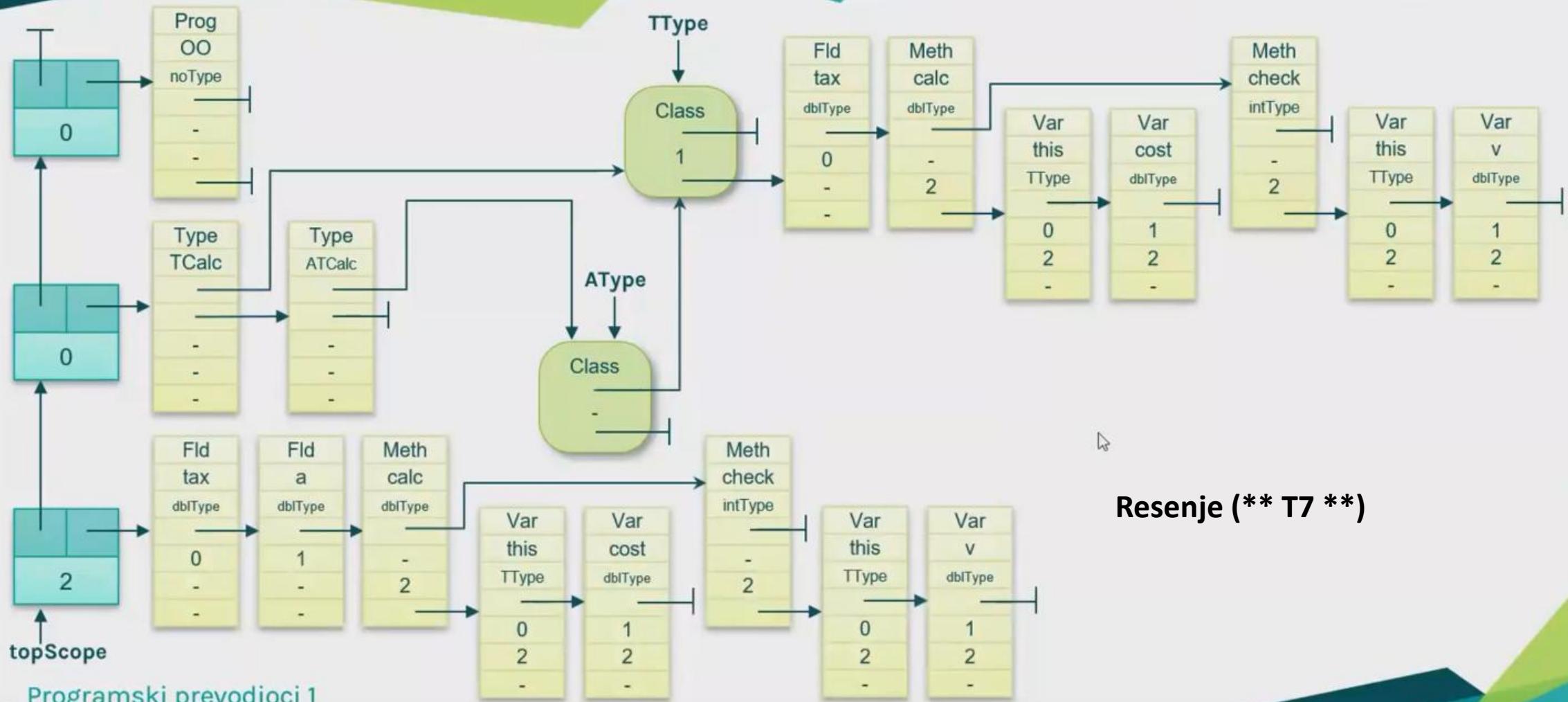
program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
{ (** T7 **)
    double calc(double c) { return ... }
} (** T8 **)
} (** T9 **)
{
void main()
    TCalc c1, c2; double cost;(** T10 **)
{
    ...
}
} (** T11 **)

```

Programski prevodioci 1

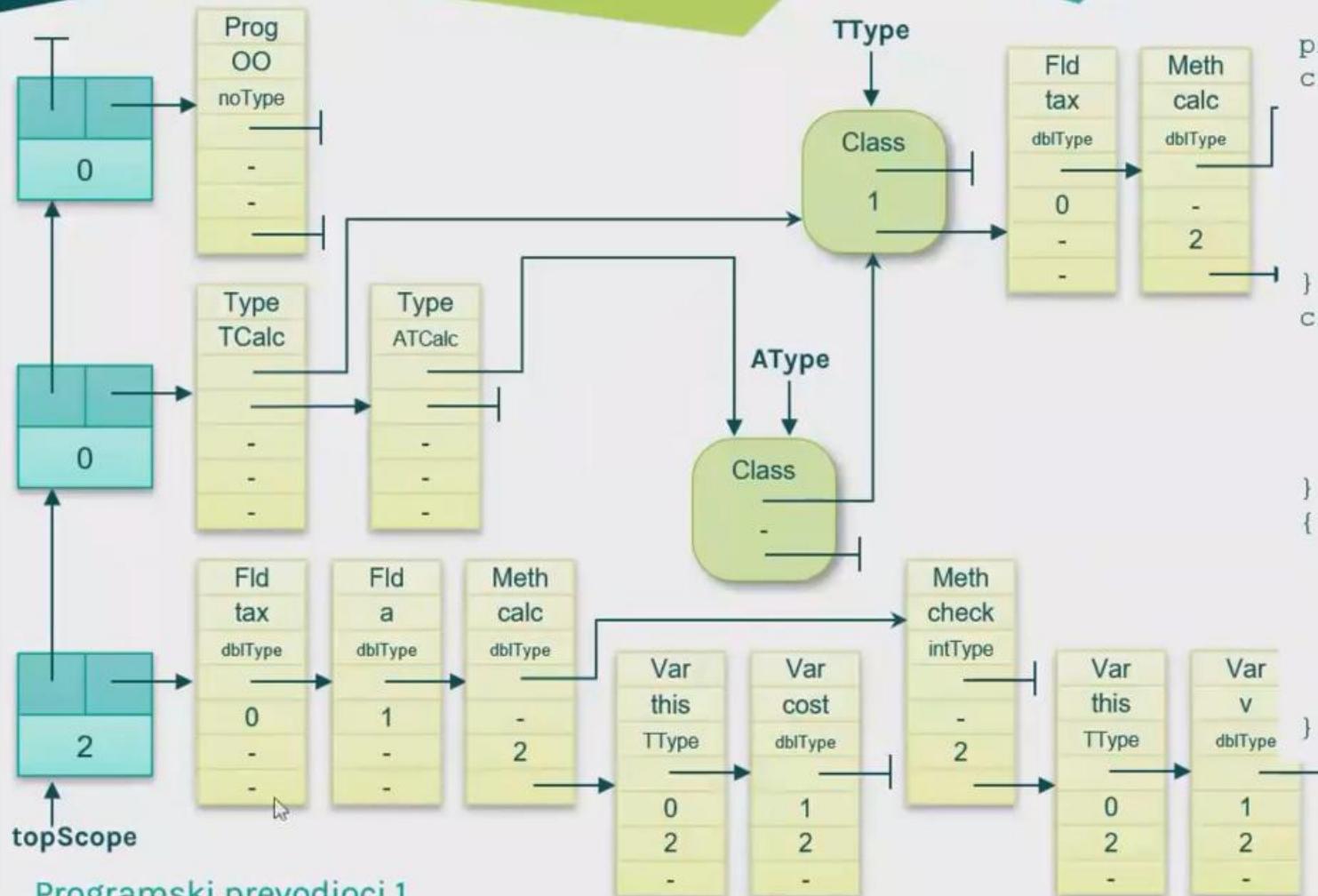


Zadatak 3.



Resenje (** T7 **)

Zadatak 3.



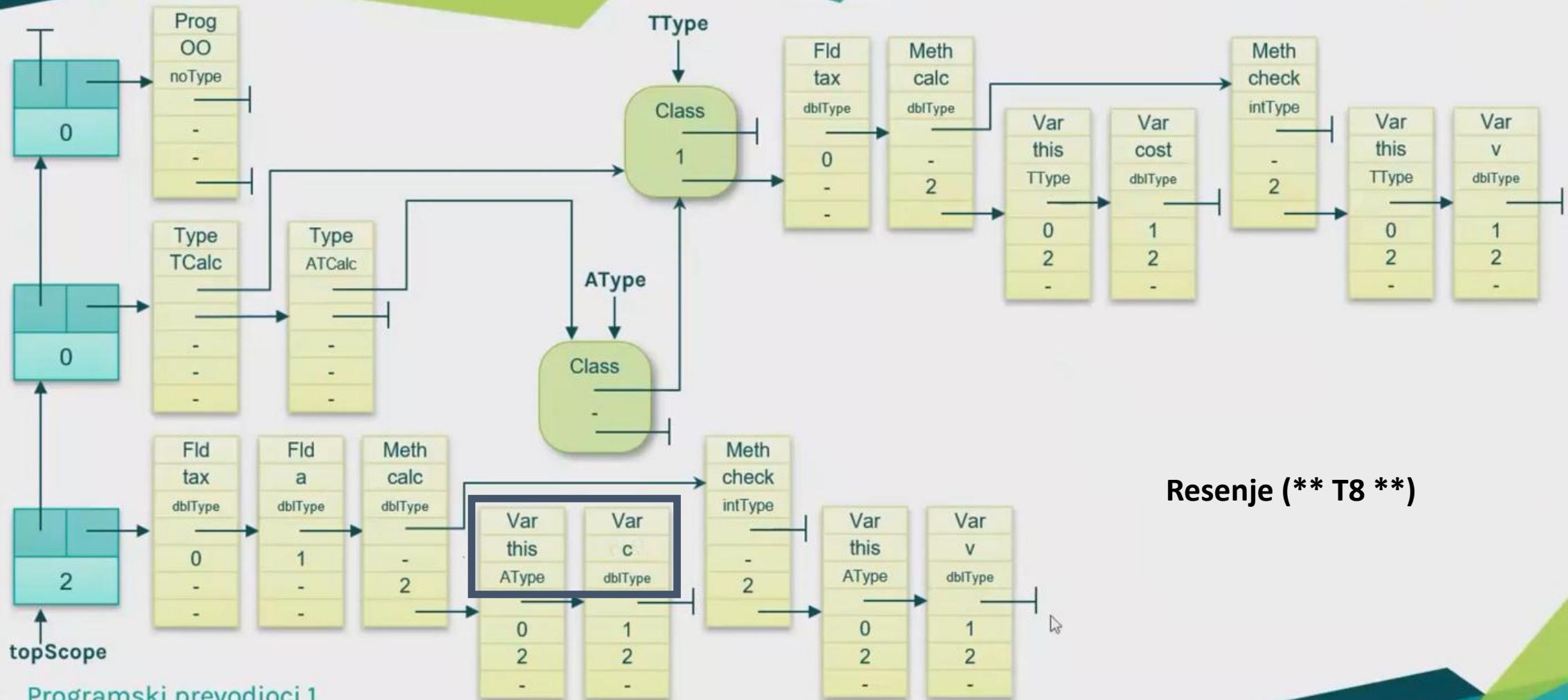
```

program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
{ (** T7 **)
    double calc(double c) { return ... }
} (** T8 **)
} (** T9 **)
{
void main()
    TCalc c1, c2; double cost;(** T10 **)
{
    ...
} (** T11 **)

```

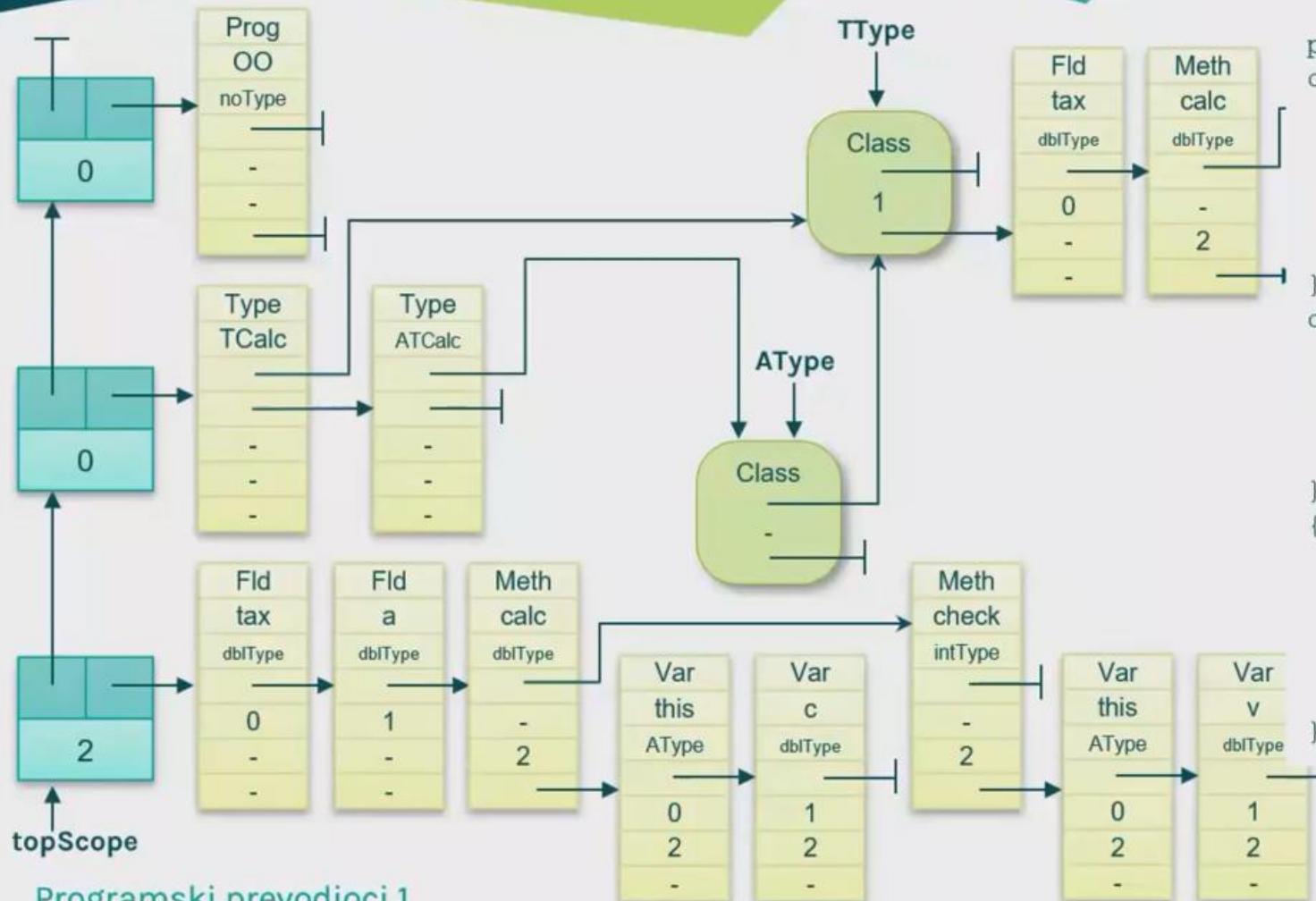
Programski prevodioci 1

Zadatak 3.



Programski prevodioci 1

Zadatak 3.



```

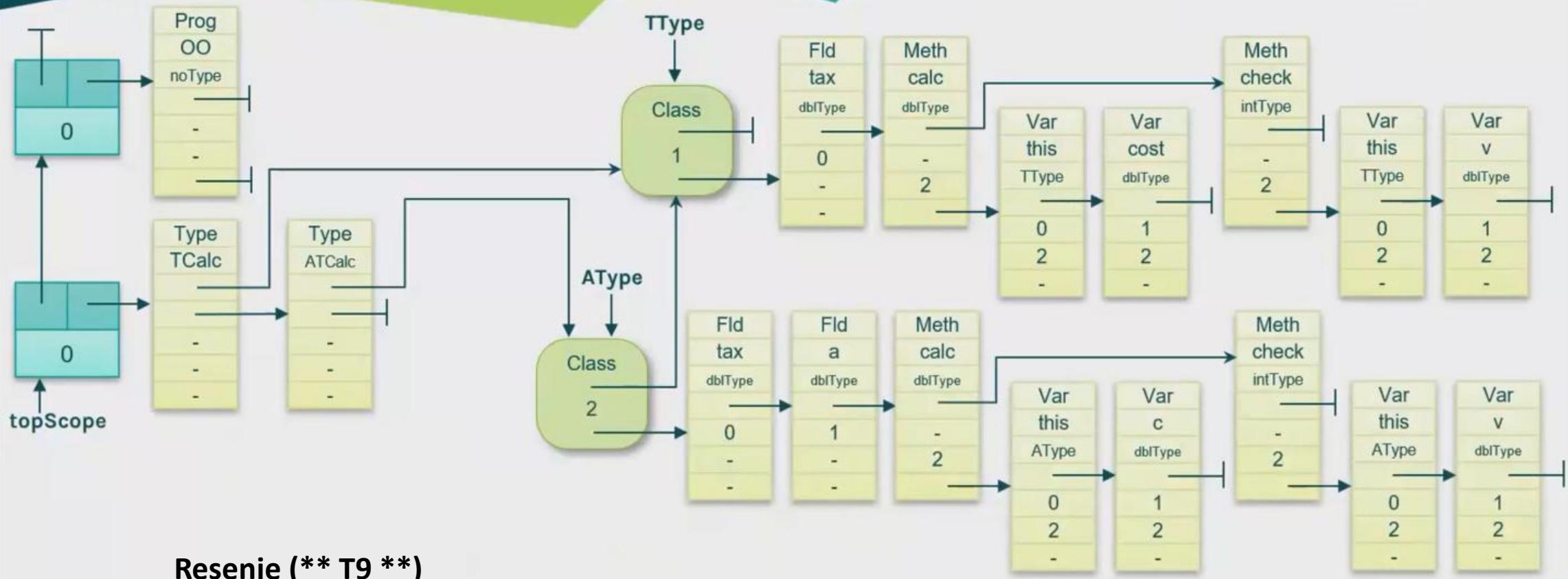
program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
{ (** T7 **)
    double calc(double c) { return ... }
} (** T8 **)
} (** T9 **)
{
void main()
    TCalc c1, c2; double cost; (** T10 **)
{
    ...
} (** T11 **)

```

Programski prevodioci 1



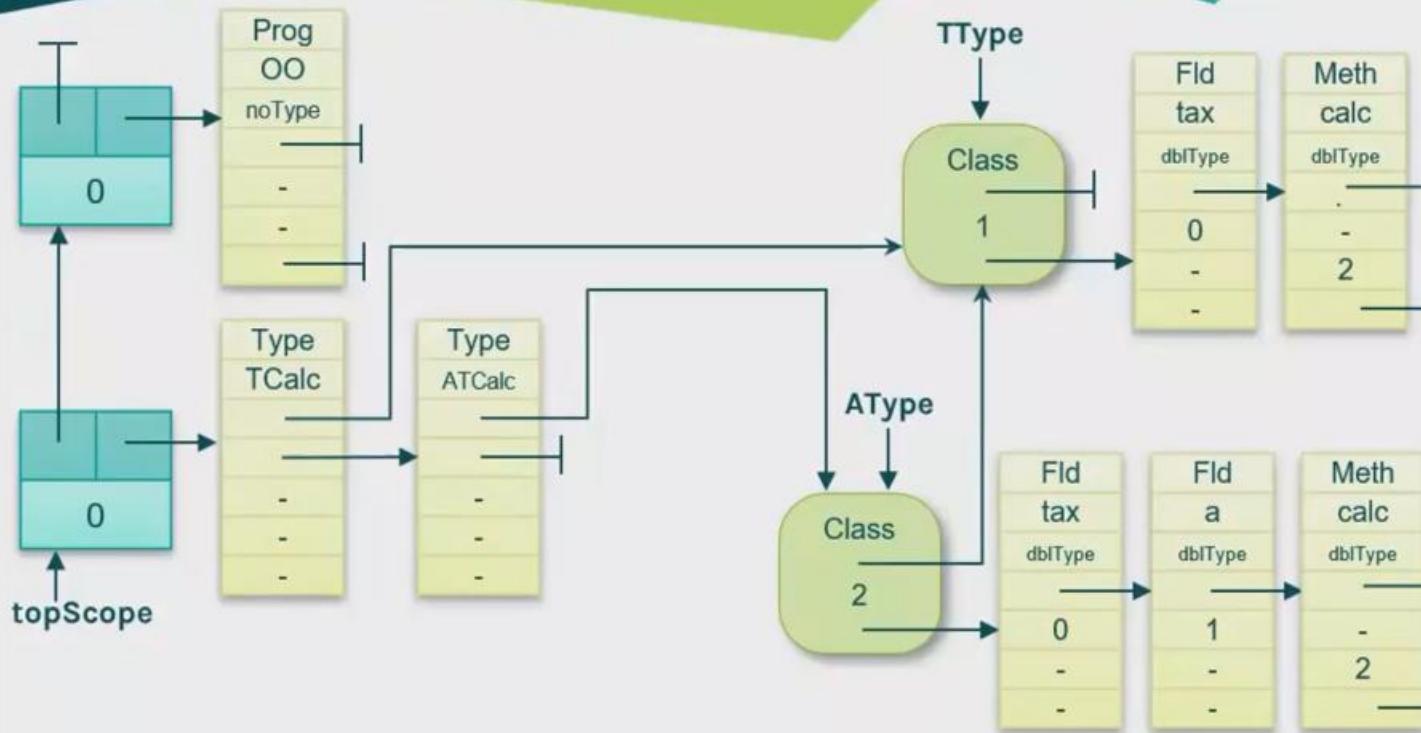
Zadatak 3.



Resenje (** T9 **)

Programski prevodioci 1

Zadatak 3.

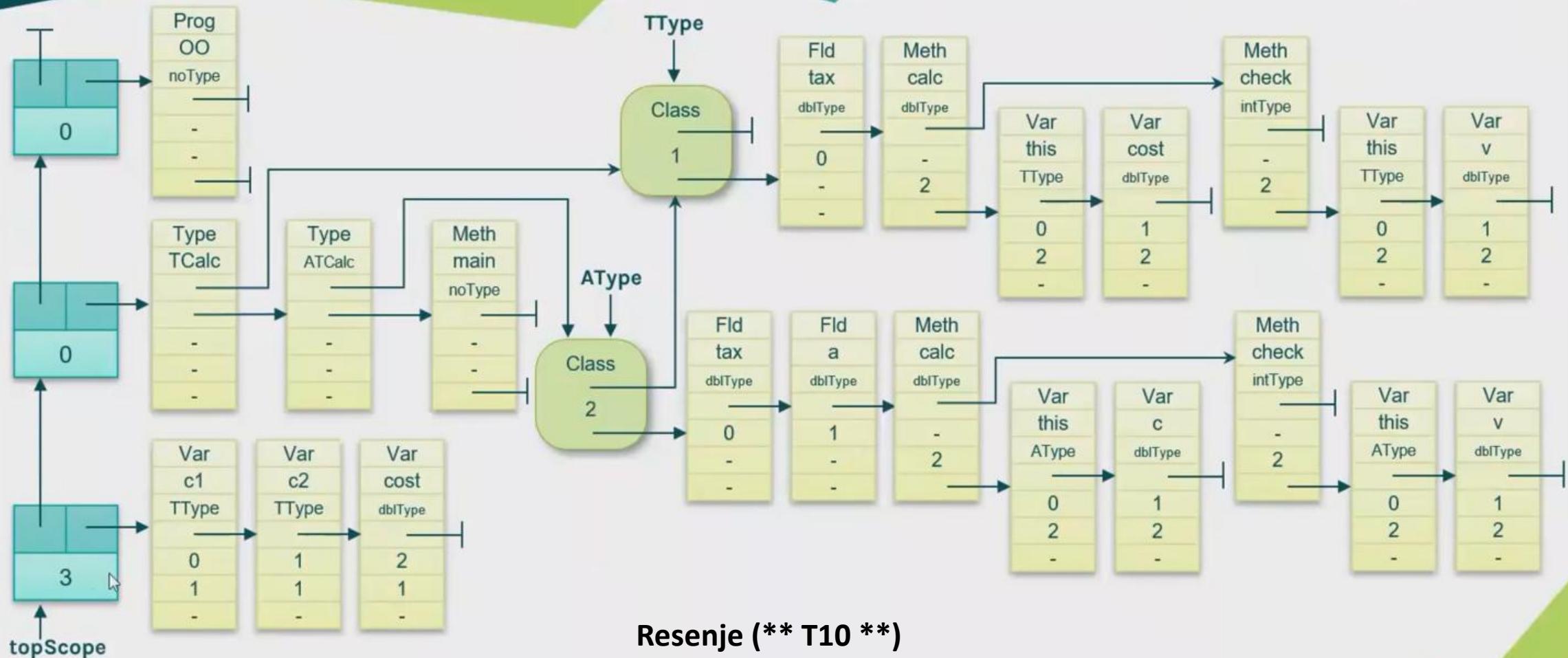


```

program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
{ (** T7 **)
    double calc(double c) { return ... }
} (** T8 **)
} (** T9 **)
{
void main()
    TCalc cl, c2; double cost;(** T10 **)
{
    ...
}
} (** T11 **)

```

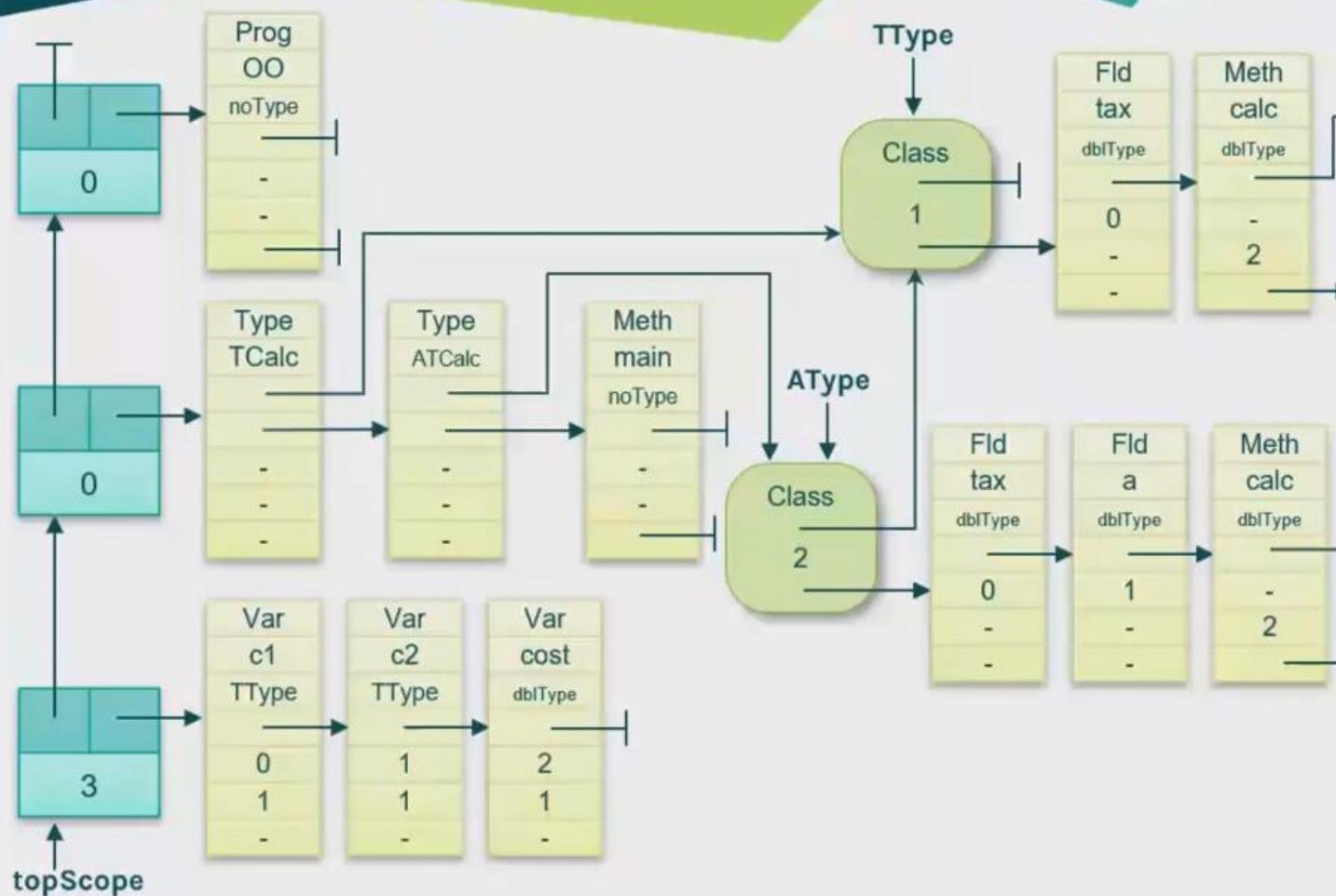
Zadatak 3.



Resenje (** T10 **)

Programski prevodioci 1

Zadatak 3.



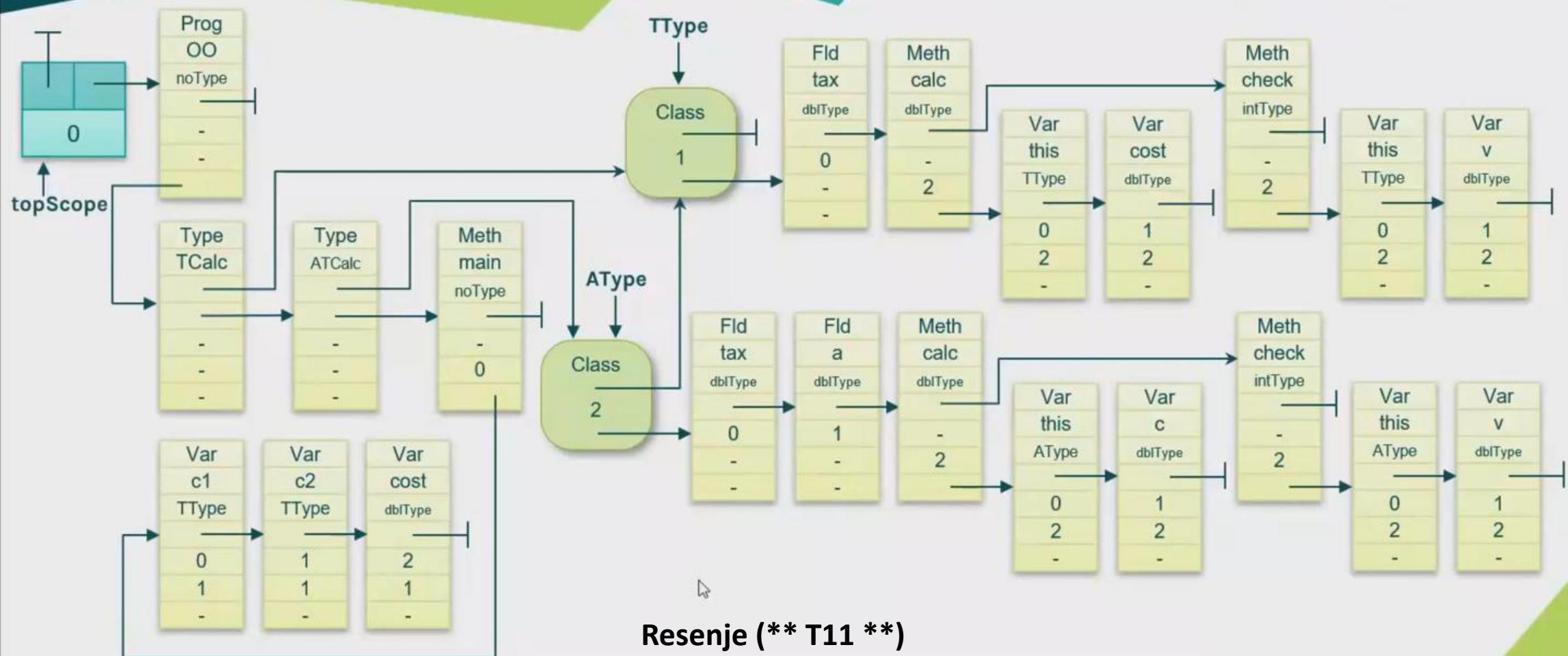
```

program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
{
    double calc(double cost){ return ... }
    int check(double v) { return ... }
} (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
{ (** T7 **)
    double calc(double c) { return ... }
} (** T8 **)
} (** T9 **)
{
    void main()
        TCalc c1, c2; double cost; (** T10 **)
    {
        ...
    } (** T11 **)
}

```

Programski prevodioci 1

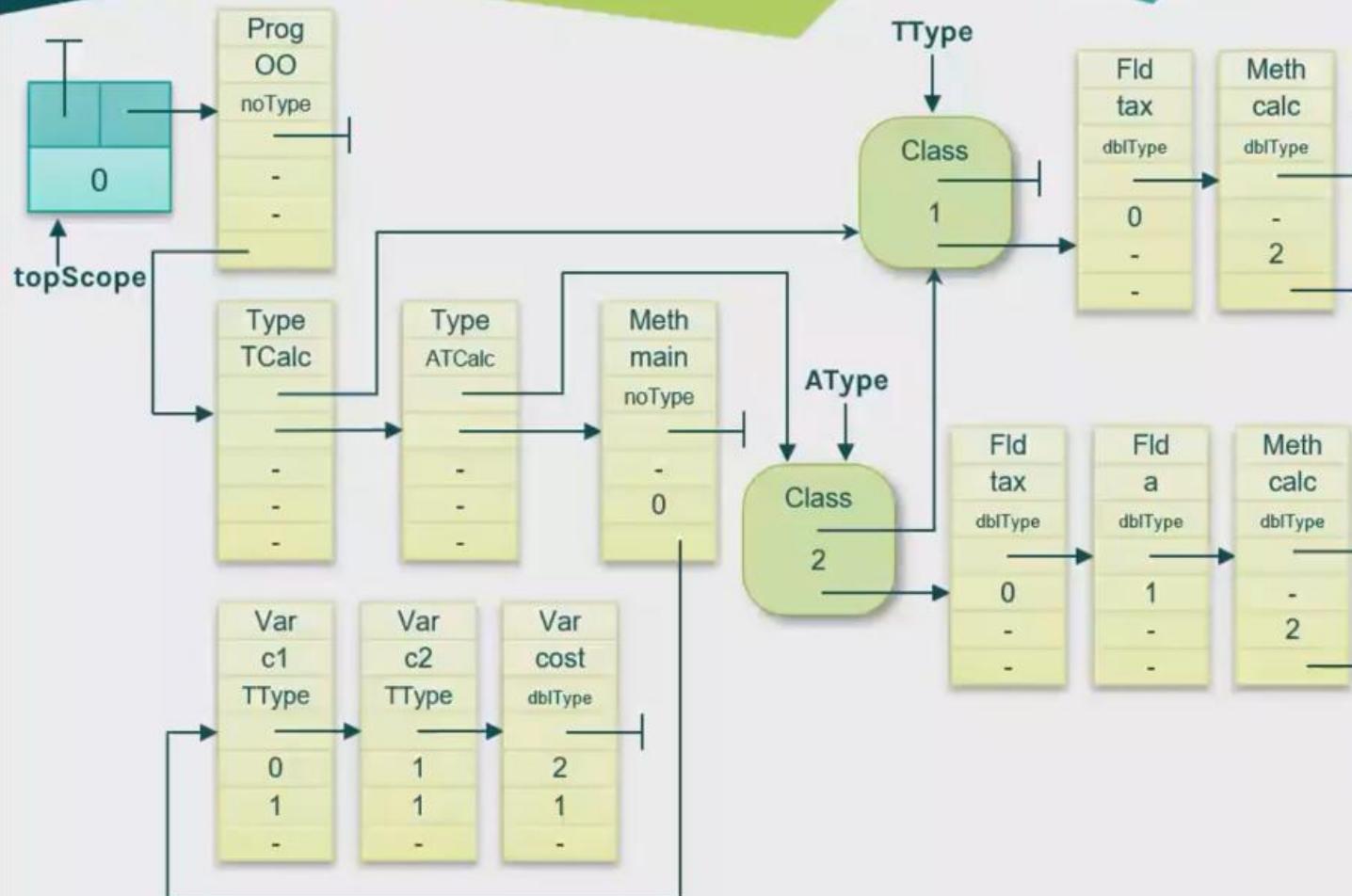
Zadatak 3.



Resenje (** T11 **)

Programski prevodioci 1

Zadatak 3.



```

program OO (** 1 **)
class TCalc (** T2 **) {
    double tax; (** T3 **)
    {
        double calc(double cost) { return ... }
        int check(double v) { return ... }
    } (** T4 **)
} (** T5 **)
class ATCalc (** T6 **) extends TCalc {
    double a;
    { (** T7 **)
        double calc(double c) { return ... }
    } (** T8 **)
} (** T9 **)
{
    void main()
        TCalc c1, c2; double cost; (** T10 **)
    {
        ...
    } (** T11 **)
}
  
```

Programski prevodioci 1